VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Agregace SIP útoků pro detekci rozsahů šiřitelů škodlivého provozu

SIP Attack Aggregations for Detection of Attack's Source Address Range

2020 Bc. Lukáš Varyš

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Lukáš Varyš**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 1801T064 Informační a komunikační bezpečnost

Téma: Agregace SIP útoků pro detekci rozsahů šiřitelů škodlivého provozu
SIP Attack Aggregations for Detection of Attack's Source Address Range

Jazyk vypracování: čeština

Zásady pro vypracování:

Bezpečnostní hrozby a zranitelnost VoIP systémů jsou v současné době aktuálním problémem. Cílem diplomové práce je implementace agregace informací o útocích na honeypot pomocí SIP protokolu. Výsledná agregace umožní detekci adresních rozsahů, z nichž útoky pocházejí, detekci DDoS útoků či charakteristických rysů jednotlivých nástrojů použitých k útokům.

Body zadání:
1. Popis metod a technologií používaných pro VoIP komunikaci a honeypoty.
2. Návrh a analýza možných agregací detekovaných útoků.
3. Implementace a otestování výkonnosti agregací.
4. Vizualizace výsledků agregace.
5. Dokumentace pro práci s výsledným řešením.

Seznam doporučené odborné literatury:

COLLIER, Mark D. a David ENDLER. Hacking exposed: unified communications & VoIP security secrets & solutions. Second edition. New York: McGraw-Hill Education, 2014. ISBN 0071798765.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Jakub Šafařík, Ph.D.**

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2020

doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 15. 5. 2020 .....

.....

Podpis studenta

Tímto bych chtěl poděkovat vedoucímu diplomové práce, panu Ing. Jakubovi Šafaříkovi Ph.D., za odborné vedení a věcné konzultace při tvorbě této diplomové práce.

Zároveň bych chtěl poděkovat mé přítelkyni a rodičům za přečtení diplomové práce a za emoční a materiální podporu při studiu.

Abstrakt

V současné době je problematika útoků na VoIP systémy stále aktuálním tématem. Na základě této problematiky byla vytvořena tato diplomová práce, jejíchž cílem je implementace agregačních algoritmů, které by mohly pomoci při detekci šiřitelů škodlivého provozu. Data pro ověření funkčnosti agregací byla dodána ze systému Beekeeper. Výsledná data z agregací jsou dále exportována do podoby vhodné pro vizualizační algoritmy. Pro práci s agregačními algoritmy a vizualizacemi byla vytvořena jednoduchá webová aplikace.

Klíčová slova: agregace, vizualizace, DDoS, VoIP, SIP

Abstract

Problematics of attacks on VoIP systems is still an actual topic. This diploma thesis was based on this problematics. The aim of this diploma thesis is implementation of aggregation algorithms which can lead to detection of malicious traffic spreaders. Data for verification of aggregation functionality was given from Beekeeper honeypot system. The result data from aggregations are exported to appropriate form for visualization algorithms. Simple web applications was developed for control over aggregation algorithms and visualizations.

Key Words: aggregation, visualization, DDoS, VoIP, SIP

Obsah

Seznam použitých zkratek a symbolů **13**

Seznam obrázků **15**

Seznam tabulek **17**

Úvod **19**

1 Technologie VoIP **21**
1.1 Signalizační protokol SIP 21
1.1.1 Architektura protokolu SIP 21
1.1.2 Adresování v protokolu SIP 22
1.1.3 Přehled SIP zpráv 22
1.1.4 Bezpečnost protokolu SIP 23
1.2 Bezpečnostní hrozby ve VoIP 24
1.2.1 Klasifikace hrozeb 24
1.3 Honeypoty ve VoIP 26
1.3.1 Honeynet 27

2 Návrh a analýza možných agregací **31**
2.1 Agregace dat v datových skladech 31
2.1.1 Bezpečnostní datové sklady 31
2.1.2 Sumarizační hierarchie datových skladů 32
2.2 Poskytnutá datová sada 33
2.3 Návrh možných agregací 34
2.4 Rozšíření stávajícího databázového modelu 35

3 Implementace, vizualizace a testování agregací **37**
3.1 Agregace dat na základě autonomních systémů 37
3.1.1 Vizualizace agregace 39
3.1.2 Výkonnostní testování agregace 43
3.2 Agregace dat na základě šiřitelů škodlivého provozu 44
3.2.1 Vizualizace agregace 45
3.2.2 Výkonnostní testování agregace 49
3.3 Agregace dat na základě dynamického útočného okna 50
3.3.1 Vizualizace agregace 51
3.3.2 Výkonnostní testování agregace 54
3.4 Agregace dat na základě signalizačního protokolu SIP 55

3.4.1 Vizualizace agregace 56
3.4.2 Výkonnostní testování agregace 59

4 Webové uživatelské rozhraní. 61

5 Závěr 63

Literatura 65

Přílohy 67

A Dokumentace pro práci s výsledným řešením 69
A.1 Backendová část 69
A.1.1 Soubor main.py 69
A.1.2 Soubor bgp_aggregation.py 69
A.1.3 Soubor data_export.py 70
A.2 Frontendová část 70
A.3 Práce s interaktivními grafy 71
A.3.1 Tlačítko exporting 71
A.3.2 Přepínání mezi měřítky 71
A.3.3 Vypnutí/zapnutí zobrazení křívek 71
A.3.4 Přiblížení na ose x 72
A.3.5 Detail na vnitřní prstenec grafu 72
A.3.6 Detail na druhou úroveň stromového grafu 72

B Obrázky 73

C Příloha v IS Edison 74

Seznam použitých zkratek a symbolů

VoIP	–	Voice over Internet Protocol
IP	–	Internet Protocol
SIP	–	Session Initiation Protocol
RTP	–	Real-time Transport Protocol
SRTP	–	Secure Real-time Transport Protocol
QoS	–	Quality of Service
UDP	–	User Datagram Protocol
UAC	–	User Agent Client
UAS	–	User Agent Server
B2BUA	–	Back To Back User Agent
URI	–	Uniform Resource Identifier
SDP	–	Session Description Protocol
HTTP	–	Hypertext Transfer Protocol
FTP	–	File Transfer Protocol
S/MIME	–	Secure/Multipurpose Internet Mail Extensions
TLS	–	Transport Layer Security
MitM	–	Man in the Middle
DoS	–	Denial of Service
DDoS	–	Distributed Denial of Service
ISO/OSI	–	Inernational Organization for Standardization / Open Systems Interconnection
IDS	–	Intrusion Detection System
GRE	–	Generic Routing Encapsulation
ASN	–	Autonomous System Number
VLSM	–	Variable-Length Subnet Mask
OS	–	Operační Systém

Seznam obrázků

1 Přehled sociálních hrozbeb (převzato z [17]) . . . 24
2 Útoky založené na odposlechu (převzato z [17]) . . . 25
3 Útoky typu MitM (převzato z [17]) . . . 25
4 Útoky za účelem úmyslného přerušení služby (převzato z [17]) . . . 25
5 Neúmyslné přerušení služby (převzato z [17]) . . . 26
6 Typy zneužití služby (převzato z [17]) . . . 26
7 Taxonomie honeynetů (převzato z [22]) . . . 28
8 Jednoduchý bezpečnostní datový sklad (převzato z [24]) . . . 32
9 Hierarchie sumarizačních tabulek (převzato z [24]) . . . 33
10 Tabulka mlp_attribute . . . 34
11 Entita BGP_info. . . . 36
12 Entita IP_info. . . . 36
13 Ukázka upravených dat ze souboru *data-add-ripe*. . . . 38
14 Ukázka dat z tabulky *bpg_info*. . . . 38
15 Ukázka dat z tabulky *ip_info*. . . . 39
16 Počet útoků z deseti nejčastějších ASN - lineární měřítko. . . . 39
17 Počet útoků z deseti nejčastějších ASN - logaritmické měřítko. . . . 40
18 Podobný průběh útoků u AS2852 a AS14061. . . . 41
19 Průběh útoků u AS199264 a AS209299. . . . 41
20 Procentuální zastoupení útoků TOP10 ASN. . . . 42
21 Počet útočných činitelů agregovaných do /8 supernetů za den 1.1.2019. . . . 43
22 Graf časové náročnosti agregace na základě autonomních systémů za období 1.1.2019-10.1.2019. . . . 44
23 Histogram počtu útoků z jednotlivých šiřitelů. . . . 46
24 Výběr oblasti z histogramu počtu útoků z jednotlivých šiřitelů. . . . 47
25 Paprskový graf využití šiřitelů škodlivého provozu. . . . 48
26 Využití šiřitelů pro AS2852. . . . 49
27 Příklad útočnách toků. . . . 50
28 Počet útočných toků z TOP 10 ASN s délkou okna 20 minut. . . . 51
29 Počet útočných toků z TOP 10 ASN s délkou okna 10 minut. . . . 51
30 Počet útočných toků z TOP 10 ASN s délkou okna 5 minut. . . . 52
31 Vnější úroveň grafu agregujícího útočné toky do subnetů a autonomních systémů. 53
32 Vnitřní úroveň grafu z Obrázku 31. . . . 53
33 Graf výkonnostního testování funkce *aggregate_attack_streams*. . . . 55
34 Vizualizace počtu SIP zpráv. . . . 57
35 Vizualizace počtu SIP zpráv - logaritmické měřítko. . . . 58
36 Útočný scénář AS2852 v období od 18.12.2018 do 26.3.2019. . . . 59

37 Výkonnostní testování funkce *aggregate_sip_messages*. 60
38 Výkonnostní testování funkce *aggregate_asn_sip_scenario*. 60
39 Příklad spuštění aplikace z terminálu. 61
40 Domovská stránka webové aplikace. 61
41 Rozbalovací lišta s odkazy na formuláře. 62
42 Rozbalovací lišta s odkazy na formuláře. 62
43 Detail na graf s tlačítkem "exporting". 71
44 Ukázka grafu s vypnutými křivkami. 72
45 Struktura MySQL databáze pro Beekeeper (převzato z [23]) 73
46 Příloha v IS Edison. 74

Seznam tabulek

1 SIP metody [2] 23
2 Třídy SIP odpovědí [2] 23
3 Data výkonnostního testování funkce *aggregate_attack_streams*. 54
4 Data výkonnostního testování funkce *aggregate_sip_messages*. 59
5 Tabulka dat výkonnostního testování funkce *aggregate_asn_sip_scenario*. 60

Úvod

Technologie VoIP je v současné době jednou z nejpopulárnějších technologií. Její největší výhody spočívají převážně v jednoduchosti nasazení a nízkých nákladech jak na nasazení, tak na provoz. Nicméně s rostoucí popularitou rostou také potenciální hrozby.

Při rostoucím množství uživatelů roste i poptávka po nových funkcionalitách a s rostoucím počtem funkcionalit může vést k nalezení nových zranitelností. V současné době jsou pro VoIP systémy největší hrozbou útoky s cílem úmyslného přerušení služby. Tyto útoky mohou pocházet z jednoho nebo více zdrojů a rovněž se mohou vyznačovat charakteristickými rysy.

Cílem této diplomové práce je vytvoření agregačních algoritmů, které umožní detekci adresních rozsahů šiřitelů, a jejich korespondujících autonomních systémů. Dále tato diplomová práce nabízí možnosti vizualizace jednotlivých agregací, včetně využití jednoduchého rozhraní pro práci s agregacemi.

První kapitola obsahuje stručný úvod do technologie VoIP. Je zde popsán signalizační protokol SIP, jeho architektura, adresování, zprávy a také bezpečnost tohoto protokolu. Dále jsou v této kapitole uvedeny bezpečnostní hrozby ve VoIP, včetně klasifikace těchto hrozeb. V poslední řadě je zde uveden stručný úvod k honeypotům ve VoIP.

Ve druhé kapitole je popsán teoretický úvod do agregací, zejména jejich využití v datových skladech. Dále jsou zde popsána data, jenž byla poskytnuta pro účely této diplomové práce a systém Beekeeper. Je zde uvoden teoretický rozbor možných agregací vzhledem k poskytnuté datové sadě a rovněž se zde zabývám možností rozšíření stávajícího databázového modelu systému Beekeper.

Třetí kapitola je zaměřena na detailní popis jednotlivých agregací a jejich implementací. Rovněž jsou zde popsány jednotlivé vizualizace, jejich přínos a případné datové anomálie, podobnosti či zajímavé datové části. Dále je zde popsáno výkonnostní testování agregací, včetně případných grafů s časovou náročností vzhledem k počtu dat a operací.

V poslední kapitole je konečně popsáno jednoduché uživatelské rozhraní v podobě webové aplikace, která může sloužit pro jednoduchou práci s agregacemi a jejich vizualizacemi.

1 Technologie VoIP

Tato kapitola slouží jako úvod do technologie Voice over IP (VoIP). Konkrétně se pak zabývá stručným úvodem do protokolu SIP a jeho využitím ve VoIP. Technologie VoIP se stala jednou z nejpoužívanějších technologií v současné době. Za svůj vzestup může především díky stále větší převaze ve využití sítí založených na přepojování paketů a protokolu IP.

VoIP využívá dvou signalizačních protokolů: SIP a H.323. Tyto protokoly jsou zodpovědné za inicializaci spojení, správu a ukončení spojení. Ačkoli je protokol H.323 i nadále hojně rozšířen, je v čím dál větší míře nahrazován jednoduchým protokolem SIP.

Zatímco protokoly SIP a H.323 slouží k managementu spojení, protokol RTP zajišťuje přenos dat v reálném čase mezi koncovými účastníky. Tento protokol však sám o sobě neposkytuje mechanismus, který by zaručil včasné doručení datového rámce, ale spoléhá na služby a protokoly nižších vrstev, aby byla splněna poskytovaná kvalita služby (QoS). [1][2][3][4]

1.1 Signalizační protokol SIP

Jedná se o kontrolní protokol aplikační vrstvy, který se stará o navázání, modifikaci a ukončení multimediálního spojení jako jsou internetová telefonní volání. Mimo navazování spojení také dokáže přizvat další účastníky do již existujících spojení, čímž se vytvářejí tzv. multicastové konference.

1.1.1 Architektura protokolu SIP

Protokol SIP je postaven na architektuře klient - server, kde mezi klientem a serverem probíhá komunikace typu žádost - odpověď. Tyto zprávy jsou obvykle přenášeny pomocí protokolu UDP.[6] Entity, využívající protokol SIP pro komunikaci, jsou nazývány *User Agents* (dále jen UA). Existují tři typy UA:

- UAC *(User Agent Client)* - klientská část, vysílá požadavky a přijímá odpovědi
- UAS *(User Agent Server)* - serverová část, přijímá požadavky a vysílá odpovědi
- B2BUA *(Back To Back User Agent)* - operuje mezi oběma koncovými body. Udržuje si informace o spojení. Jedná se o bránu mezi účastníky.

Dále se pak dle [2] architektura skládá ze serverové části, která se dá logicky rozdělit do několika částí:

- Proxy server - tváří se jako server i jako klient za účelem sestavení hovorů mezi UAC. Směruje požadavky, které obdržel od ostatních entit, a pokud je třeba, může je i pozměnit.
- Redirect server - umožňuje přesměrování klientů na alternativní SIP URI.

- Registrar server - zpracovává žádosti o registraci SIP klientů a uchává informace o těchto žádostech.
- Location server - někdy také *location service*. Často bývá součástí registrar serveru. Udržuje informace o umístění SIP klientů a SIP proxy serverů.

Toto rozdělení je však pouze konceptuální. V praxi se mohou objevovat řešení obsahující všechny časti v jednom serveru obecně nazývaném SIP server. Nicméně také se můžeme setkat s řešením, kde jsou jednotlivé serverové části rozdělené z důvodů jako jsou např. škálovatelnost, redundance nebo samostatné zpracování požadavků.

1.1.2 Adresování v protokolu SIP

Entity v SIP rozeznávájí uživatele na základě jeho SIP URI definovaného v RFC 2396.[7] Formát SIP URI je velmi podobný formátu SMTP, sestávající se z uživatele, domény a oddělovacího znaku @. Obecná forma SIP URI dle RFC 3261 je následující:

```
sip:user:password@host:port;uri-parameters?headers
```

Účel jednotlivých parametrů vyplývá z jejich názvu. Pokud je to možné, dosadí se za nevyplněné parametry výchozí hodnoty, nicméně některé části je lepší z očividných důvodů nespecifikovat (např. *password*).

1.1.3 Přehled SIP zpráv

Protokol SIP je textově orientovaný protokol s velmi podobnou sémantikou jako má protokol HTTP.[8] V protokolu SIP jsou definovány dva typy zpráv:

- žádosti (nebo také metody)
- odpovědi (nebo také stavové kódy)

Oba typy zpráv se skládají z hlavičky a těla zprávy. Hlavička má přesně definovanou strukturu, která se liší pro žádosti a odpovědi.

1.1.3.1 Žádosti Žádosti slouží k zahájení výměny zpráv mezi SIP entitami. Pomocí těchto žádostí mohou spolu dvě entity navázat, kontrolovat a ukončovat spojení. Nejdůležitější část SIP žádosti je první řádek, kde je specifikováno klíčové slovo, které specifikuje typ žádosti, viz tabulka 1.

SIP žádost	Účel
REGISTER	Registrace SIP klienta
INVITE	Zahájení spojení mezi dvěma klienty
ACK	Potvrzení navázání spojení
BYE	Ukončení spojení
CANCEL	Zrušení požadavku
OPTIONS	Žádost o informace o možnostech serveru

Tabulka 1: SIP metody [2]

1.1.3.2 Odpovědi Po přijetí žádosti musí adresát žádosti odpovědět zdroji žádosti. K tomu slouží tzv. stavové kódy. Tyto kódy se dělí do několi tříd, jenž každá obsahuje jiné typy odpovědí. První číslo stavového kódu definuje třídu odpovědi, viz tabulka 2 .

Stavový kód	Třída
1xx	Provizorní odpověď
2xx	Úspěch
3xx	Přesměrování
4xx	Chyba klienta
5xx	Chyba serveru
6xx	Globální chyba

Tabulka 2: Třídy SIP odpovědí [2]

1.1.4 Bezpečnost protokolu SIP

SIP se stal dominantním signalizačním protokolem v technologii VoIP hlavně díky své jednoduchosti. Nicméně se díky jeho jednoduchosti a textové podobě stal také více náchylným k bezpečnostním hrozbám.

Jak již bylo zmíněno dříve, protokol SIP je postaven na protokolu IP. Z toho vyplývá, že SIP musí zahrnovat bezpečnostní mechanismy obdobné dalším protokolům využívajících protokolu IP, jako jsou např. HTTP nebo FTP [9].

Ve většině případů je SIP využíván pro správu VoIP spojení, kde je nezbytné zabezpečení signalizační a komunikační části. Jelikož je SIP zodpovědný pouze za kontrolu multimédiálních sezení, je třeba využití externích protokolů. Jedním z takových protokolů je SRTP [10].

Zabezpečení signalizační části je integrováno přímo ve specifikaci protokolu SIP. Jedná se o mechanismus zajišťující autentizaci uživatelů a důvěryhodnost při přenosu dat, tzv. S/MIME.

Mezi další podpůrné bezpečnostní protokoly patří TLS [11]. Tento protokol je využíván pro bezpečný přenos zpráv mezi SIP entitami. Pokud se SIP zprávy zašifrují pomocí S/MIME, znepřístupní se tak citlivá data i pro SIP proxy servery [13]. Nicméně některá důležitá pole hlavičky

SIP zpráv nesmí být šifrována z důvodů správného směrování zpráv. K těmto hlavičkovým polím patří např. *To*, *From*, *Call-ID*, *CSeq* a *Contact*.

1.2 Bezpečnostní hrozby ve VoIP

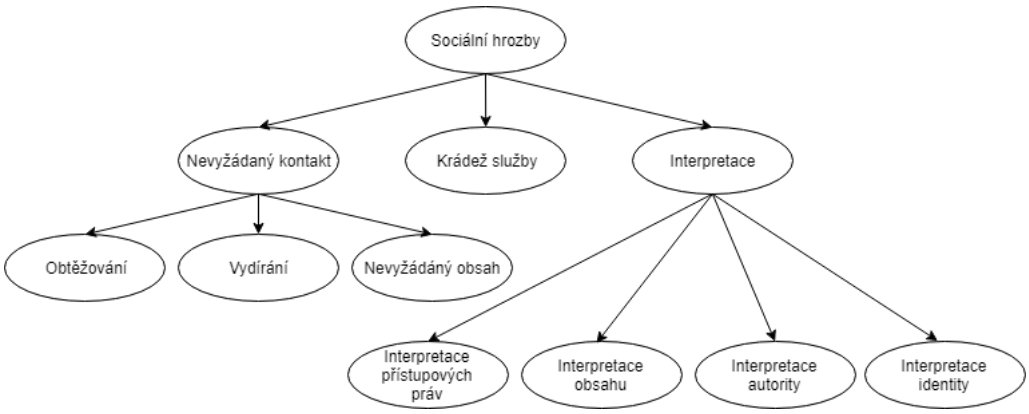
Tato kapitola slouží jako stručný popis a úvod do problematiky bezpečnosti technologie VoIP. V počátcích nebyla bezpečnost ve VoIP tak velkým tématem jako samotné využití této nové technologie. Poskytovatelé služeb, komunita a veřejnost byli více zaujati cenou, funkčností a spolehlivostí služby. Nicméně s rostoucí popularitou a využitím začala být bezpečnost VoIP závažným problémem, jako u mnohých dalších moderních služeb a technologií.

V následujících podkapitolách se zabývám základní klasifikací hrozeb pro VoIP systémy na základě dle Keromytise A. [14] a VoIP Security Aliance [15]. Tato základní klasifikace pomháhá při identifikaci útoků a tím může zjednodušit správnou volbu nových bezpečnostních mechanismů. Na tuto základní klasifikaci bychom se mohli dívat i z pohledu důvěryhodnosti, integrity a dostupnosti tak, jak je popsána v [16]. Ačkoli klasifikace podle Xin J. [16] přidává další úroveň v klasifikační hierarchii, z popisu uvedeného v literatuře [14] a [15] jasně vyplývá, zdali se jedná o hrozbu zasahující důvěryhodnost, integritu nebo dostupnost služby.

1.2.1 Klasifikace hrozeb

Jak již bylo zmíněno v 1.2, tato podkapitola slouží k základní klasifikaci bezpečnostních hrozeb VoIP na základě klasifikace dle Keromytise A. a VoIP Security Aliance [14][15]. Tato taxonomie definuje bezpečnostní hrozby vůči nasazení VoIP, službám a koncovým uživatelům. Klíčovými elementy taxonomie jsou sociální hrozby, odposlouchávání, zachycení a modifikace zpráv, úmyslné přerušení služby, neúmyslné přerušení služby, zneužití služby a fyzické hrozby.

1.2.1.1 Sociální hrozby Tyto hrozby se zaměřují na manipulaci sociálního kontextu mezi komunikujícími stranami. Útočník se může jevit jako důveryhodná entita a předávat falešné informace cílenému uživateli. Sociální hrozby lze dále dělit dle Obrázku 1.



Obrázek 1: Přehled sociálních hrozbeb (převzato z [17])

1.2.1.2 Odposlouchávání

K odposlechu dochází pokud útočník zachytí datový tok mezi dvěma nebo více uživateli a obsah datového toku zůstane nezměněn. Útočník má však přístup k informacím, které mezi sebou sdílí všechny zúčastněné strany.

Odposlech může velmi rychle přejít v závažný problém, převážně pokud jsou posílána data přes nezabezpečenou, nebo špatně zabezpečenou síť. Avšak s použitím moderních šifrovacích metod a řádně zabezpečené sítě je dopad tohoto útoku minimální. Odposlouchávání lze dále dělit dle Obrázku 2.[18]

Obrázek 2: Útoky založené na odposlechu (převzato z [17])

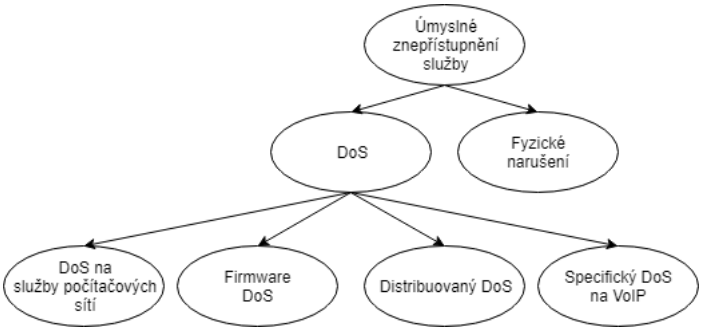
1.2.1.3 Zachycení a modifikace zpráv

Hrozby z této kategorie pojednávají o útocích, kde se útočník snaží zachytit a pozměnit data mezi dvěma nebo více koncovými body. Tento typ hrozby je rovněž znám jako MitM. Zachycení a modifikace zprávy lze dále dělit dle Obrázku 3.

Obrázek 3: Útoky typu MitM (převzato z [17])

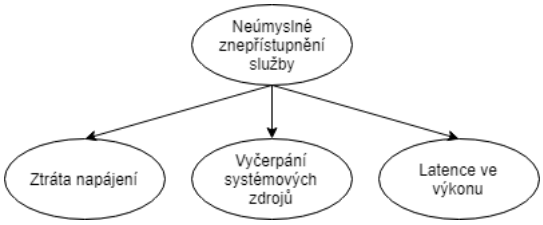
1.2.1.4 Úmyslné přerušení služby

Veškeré hrozby z této kategorie se zaměřují na přerušení přístupu uživatelům k VoIP nebo k jiné službě. Ve většině případů nemá útočník osobní zisk z útoku, avšak motivací útočníka může být více. Úmyslné přerušení služby lze dále dělit dle Obrázku 4.



Obrázek 4: Útoky za účelem úmyslného přerušení služby (převzato z [17])

1.2.1.5 Neúmyslné přerušení služby Jedná se o problémy, které mohou neúmyslně způsobit nestabilitu nebo nedostupnost VoIP služeb. Mezi tyto hrozby se řadí ztráta napájení, vyčerpání systémových zdrojů kvůli příliš velkému počtu připojených uživatelů nebo problémy s výkonem služby, které můžou zapříčinit degradaci kvality hovoru viz Obrázek 5.[14][17]



Obrázek 5: Neúmyslné přerušení služby (převzato z [17])

1.2.1.6 Zneužití služby Tyto hrozby pokrývají veškeré hrozby týkající se podvodných aktivit ve VoIP. Příkladem takového zneužití jsou podvody s poplatky či přesměrování na prémiová čísla (viz Obrázek 6).

Obrázek 6: Typy zneužití služby (převzato z [17])

1.2.1.7 Fyzické hrozby Odkazují na neautorizovaný fyzický přístup k VoIP zařízením nebo fyzické vrstvě sítě podle referečního modelu ISO/OSI. Z tohoto důvodu by měl být fyzický přístup k zařízením jen autorizovaným osobám a zařízení by měla být bezpečně uchovávána v prostředí s omezeným přístupem.

1.3 Honeypoty ve VoIP

Honeypot může být popsán jako počítačové zařízení, které je důkladně monitorováno za účelem sběru informací o aktivitách přilákaného útočníka [19]. Avšak honeypoty nemusí být vždy plnohodnotné počítačové zařízení nebo systém. Může se jednat pouze o omezenou část systému, jako je *chroot* na Unix-like operačních systémech. Podle míry interakce s útočníkem se honeypoty dělí na:

- Honeypoty s nízkou mírou interakce
- Honeypoty s vysokou mírou interakce

Honeypoty s nízkou mírou interakce neposkytují reálné přostředí, ale simulují operační systémy a jím poskytované služby. Tento typ honeypotu neposkytuje velké množství informací vzhledem k omezení pouze na emulované prostředí. Útočník nemá takové možnosti a tudíž je míra zneužití těchto honeypotů poměrné malá.

Druhým typem jsou honeypoty s vysokou mírou interakce, které poskytují reálné prostředí, operační systémy a aplikace, kde se útočníci mohou pohybovat svobodněji, čímž přispívají k většímu množství zachycených dat. [21]

Jednou z dalších možností, jak dělit honeypoty, je rozdělení podle účelu použití. Takto se tedy honeypoty dělá na:

- Virtuální honeypoty - poskytují kompletní virtuální operační systém. Díky možnostem virtualizace je možné realizovat celý honeynet na jednom, nebo pár fyzických zařízení.
- Bezdrátové honeypoty - cílem těchto honeypotů je ochrana bezdrátových sítí. Mohou poskytovat stovky fiktivních sítí a falešnou komunikaci mezi klienty sítě.
- Honeypoty založené na vyhledávačích/webu - předstírají internetové aplikace, ze kterých máme možnost obdržet nové vzorky malware či exploitů.
- Klientské honeypoty - jedná se o fiktivní koncové spoje, o které se útočníci mohou zajímat z nejrůznějších důvodů, typicky připojení do botnetu nebo výpočetního clusteru.
- Farmy honeypotů - tyto farmy slouží k zjednodušení zprovoznění velkého honeynetu. Honeypoty jsou koncetrovány na jednom místě, na které je útočník přesměrován.

1.3.1 Honeynet

Jedná se o infrastrukturu honeypotů, která simuluje celou síťovou topologii s vybranými službami.[20] Tímto se vytváří prostředí velmi podobné produkci. S větší škálou použitých technologií a služeb se otevírá více možností pro potencionální napadení a tím i větší počet dat k analýze a odhalení útočných technik. Honeynet by měl zastávat tři hlavní funkce:

- kontrolovat data,
- zachytávat data,
- uchovávat data.

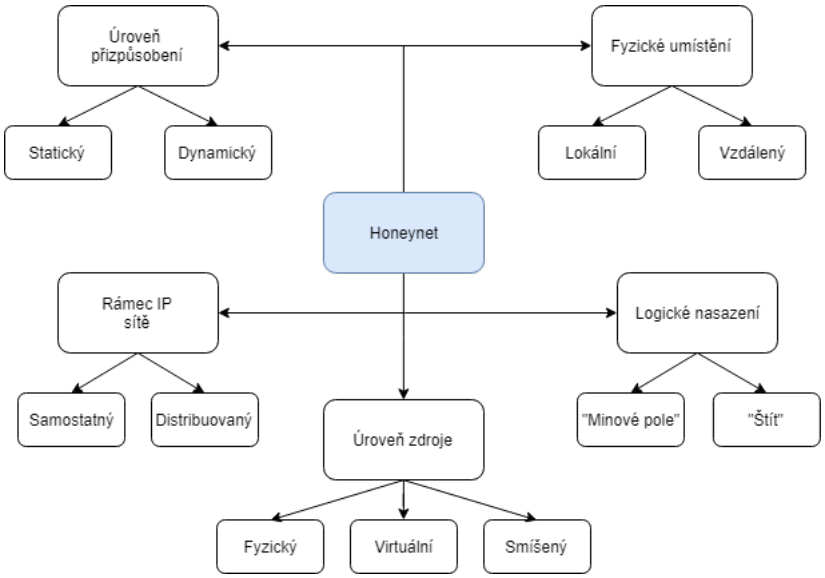
Kontrola dat je souborem opatření ke snížení rizika, které představuje útočník, tak, aby se kompromitovaný honeypot nestal odrazovým můstkem pro útok na ostatní/produkční zařízení. Proto je třeba kontrolovat útoky za účelem obrany ostatních zařízení a systémů mimo honeynet.

Smyslem zachytávání dat je logování veškeré útočníkovy aktivity pro další investigaci. Jsou definované tři kritické vrstvy zachytávání dat: logy na firewallu (vstupní a výstupní spojení),

síťový provoz (každý vstupní/výstupní paket z honeynetu a jeho obsah) a aktivita systému (systémová volání, změny v souber, atd.).

Úchováváním dat jsou pak myšleny veškeré prostředky pro bezpečné přenesení zachycených dat z distribuovaného systému do bezpečného centralizovaného úložného bodu. Z toho důvodu, že honeypoty jsou samy o sobě nezabezpečené systémy, musí být zachycená data přesunuta na bezpečné úložiště co nejdříve.

Na Obrázku 7 je zobrazeno klasifikační schéma architektury honeynetu tak, jak bylo popsáno v literatuře.[22] Uvedené schéma popisuje pět pilířů návrhu honeynetu: úroveň zdroje, úroveň přizpůsobení, fyzické umístění, rámec IP sítě a logické nasazení.



Obrázek 7: Taxonomie honeynetů (převzato z [22])

1.3.1.1 Úroveň zdroje Je kritérium použité k rozdělení honeynetu na základě platformy, na které honeynet běží. Dělí se na *fyzické*, *virtuální* a *smíšené* (někdy také *hybridní*).

Fyzické honeynety se skládají z několika honeypotů běžících na fyzickém stroji, tvořící určitou síťovou topologii.

Virtuální honeynety jsou tvořeny virtuálními honeypoty, které jsou hostovány na jednom nebo více fyzických strojích. Tyto honeynety se dají dál dělit na samostatné a hybridní virtuální honeynety.

Smíšené honeynety jsou složeny jak z virtuálních, tak z fyzických honeypotů. Tento typ honeynetu přináší rovnováhu mezi efektivním využitím zdrojů a přesností služeb.

1.3.1.2 Úroveň přizpůsobení Tato úroveň odkazuje na možnosti dynamického upravování konfigurace a topologie honeynetu. Z tohoto ohledu může být honeynet *statický* nebo *dynamický*.

Statický honeynet je takový honeynet, který nemůže být po nasazení změněn nebo překonfigurován. Největší nevýhoda statického honeynetu spočívá v tom, že pokud by byla potřeba překonfigurovat jeden z honeypotů, musí se znovu nasadit celý honeynet.

Oproti tomu u dynamických honeynetů můžeme měnit konfiguraci nasazených honeypotů, jejich topologii nebo je odstranit, či přidat další honeypoty. Tímto může administrátor měnit nastavení tak, aby mohl reagovat na události vyvolané např. IDS.

1.3.1.3 Rámec IP sítě Síťový rámec udává, jakým způsobem jsou v honeynetu přiřazeny IP adresy jednotlivým honeypotům. Tento rámec můžeme klasifikovat do dvou kategorií: *samostatný* a *distribuovaný*.

V samostatném honeynetu mají všechny honeypoty IP adresy z jednoho společného síťového prefixu a sdílejí jednotný nástroj pro zachycení veškerých dat z celého honeynetu.

Distribuovaný honeynet pokrývá více sítí najednou z důvodu poskytnutí větších možností pro zachycení podezřelých síťových událostí. Nasazení tohoto typu honeynetu je vhodné pro organizace, které využívají více než jednoho honeynetu v jejich distribuovaném prostředí.

1.3.1.4 Fyzické umístění Z hlediska fyzického umístění se honeynety dělí na *lokální*, nebo *vzdálené*.

Lokální honeynet je síť honeypotů fyzicky umístěna v laboratoři, kanceláři nebo budově organizace.

U vzdáleného honeynetu není fyzické umístění překážkou. Honeynet nemusí být fyzicky postaven celý na jednom místě. Jednotlivé honeypoty mohou být roztroušeny v různých lokacích. Takto roztroušené honeypoty mohou být integrovány do jednoho honeynetu pomocí tunelovacích technologií, jako je GRE.

1.3.1.5 Logické nasazení O logickém nasazení se uvažuje jako o logickém vztahu mezi honeynetem a produkční sítí. Tato strategie může být klasifikována do dvou kategorií: tzv. *"minové pole"* a *"štít"*.

"Minové pole"je obdobnou implementací pozemních min. Honeynety využívající tuto strategii vyčkávají a pasivně začnou se sběrem dat po interakci. Honeypoty jsou často logicky nasazeny v produkční sítí čili jsou schopné zpracovávat jak produční, tak škodlivý provoz. Takovým honeypotům jsou pak přiřazený nevyužité IP adresy z adresního prostoru produkční sítě.

Při nasazení typu "štít"je honeynet nasazen jako zrcadlo produkční sítě. Tato strategie dovoluje IDS a ADS investigaci síťového provozu na základě cílových portů. Pokud je provoz zajímavý, je přeposlán do štítu, čímž brání skutečný systém před útokem. Zároveň je možné analyzovat útočníkovo chování.

2 Návrh a analýza možných agregací

V předešelých kapitolách je uveden základ do problematiky technologie VoIP, bezpečnosti technologie VoIP a agregace dat. Dalším krokem pro vypracování agregačních metod je analýza poskytnuté datové sady, na které se budou agregace provádět. Na základě této analýzy lze začít uvažovat o případném rozšíření databázového modelu, nebo využití jeho stávajících entit.

2.1 Agregace dat v datových skladech

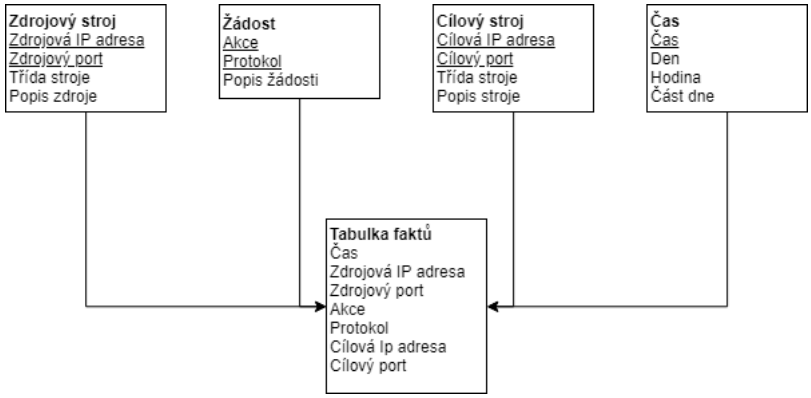
Agregace dat je proces, ve kterém jsou informace shromážděny a vyjádřeny v sumární podobě. Obecným cílem agregace je získání více informací o konkrétní skupině, založené na specifické proměnné. Takovou proměnnou může být věk, profese nebo příjem. Vzhledem k této diplomové práci se může jednat o zdroj útoku, subnet nebo typ SIP zprávy.

2.1.1 Bezpečnostní datové sklady

V literatuře [24] je popsán jednoduchý model bezpečnostního datového skladu, který je zobrazen na Obrázku 8. Obecně lze datové sklady považovat za relační databázi, která slouží pro uchovávání rozsáhlých souborů dat, jenž se využívají pro analytické zpracování.

Datové sklady se mohou využívat také pro uskladnění událostí, které souvisejí s bezpečností počítačových systémů a sítí. Tyto události jsou následně analyzovány bezpečnostními experty, a mohou sloužit jako styčný bod pro návrh a nasazení nových bezpečnostních opatření.

Kyberbezpečnostní datové sklady mohou být tvořeny rozličnými síťovými daty. Pro specifická síťová data mohou být navrženy specifické datové sklady. Jednoduchým příkladem takového datového skladu může být ukládání logů z firewallu, viz Obrázek 8. Tento jednoduchý model zobrazuje čtyři hlavní komponenty struktury *"Kdo-Co-Komu-Kdy"* do čtyř tabulek *zdroj útoku*, *žádost*, *cíl útoku* a *čas*. Neméně důležitou tabulkou tohoto jednoduchého datového skladu je pak tabulka faktů (nebo také událostní tabulka). Tato tabulka obsahuje atributy z ostatních tabulek jako odvozené klíče. Tyto atributy pak dovolují přístup a seskupování událostí na základě hodnot z ostatních tabulek.

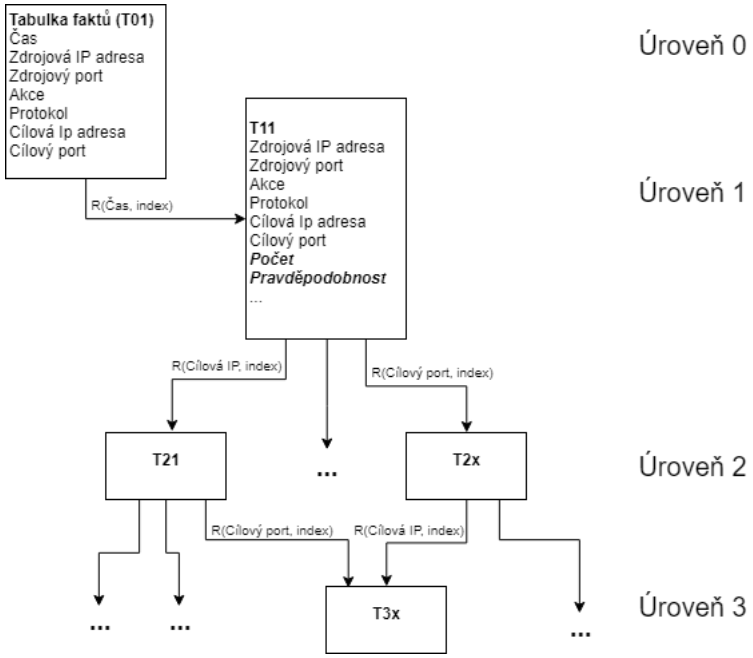


Obrázek 8: Jednoduchý bezpečnostní datový sklad (převzato z [24])

2.1.2 Sumarizační hierarchie datových skladů

Jednou z fundamentálních operací datových skladů je zpracování tabulky faktů. Díky agregacím se dají z tabulky faktů vytvořit nové, agregované tabulky, které obsahují informace o událostech, a sumarizují informace pro každou agregaci. Tyto nové tabulky se nazývají sumarizační tabulky. Návrh hierarchie sumarizačních tabulek je velmi důležitý, jelikož přináší systematickou metodu přístupu k agregovaným informacím. Sumarizační tabulky tak přispívají k různorodým analytickým a vizualizačním přístupům, které vedou k jednoduššímu pochopení bezpečnostních událostí.

Hierarchie sumarizačních tabulek může být modelována jako orientovaný acyklický graf. Příklad na Obrázku 9 zobrazuje možnou hierarchii sumarizačních tabulek z již uvedeného datového skladu z Obrázku 8.



Obrázek 9: Hierarchie sumarizačních tabulek (převzato z [24])

2.2 Poskytnutá datová sada

Pro potřeby této diplomové práce byla poskytnuta data z informačního systému Beekeeper [23]. Tento systém byl původně pro skladování dat a autonomní klasifikaci SIP útoků. Avšak postupem času se do systému začaly přidávat další funkce. Hlavní funkce systému Beekeeper jsou:

- datový sklad
- agregátor a čistič dat
- klasifikátor útoků
- monitoring a správa sond
- analyzátor klasifikovaných dat
- zdroj dat pro další systémy

Poskytnutá však nebyla data z celé databáze systému Beekeeper, ale jednalo se o data z období od *18.12.2018* do *26.3.2019* z tabulky *mlp_attribute*. Tato entita obsahuje sloupce: identifikátor útoku, IP adresa a port útočníka, transportní protokol, počet spojení, počet jednotlivých SIP zpráv (register, invite, ack, bye, cancel, options, subscribe), množství zpráv na spojení, časová známka začátku útoku, délka trvání útoku, klasifikace, informace o lokalitě útočníka (stát, region, město, časová známka geolokace). Na Obrázku 10 je tabulka mlp_attribute. Celá databáze systému Beekeeper je k náhledu na Obrázku 45.

Field	Type
id	int(10) unsigned
source	int(10) unsigned
ip	varchar(40)
port	int(10) unsigned
transport	enum('udp','tcp','tls','')
connectionCount	int(10) unsigned
regCount	int(10) unsigned
invCount	int(10) unsigned
ackCount	int(10) unsigned
byeCount	int(10) unsigned
canCount	int(10) unsigned
optCount	int(10) unsigned
subCount	int(10) unsigned
connectionRate	float unsigned
connectionRoot	int(10) unsigned
started	datetime
delta	float
result	enum('','opt_test','opt_scan','call_test','ukwSIP/noSIP','reg&call','reg_test','reg_test_high','reg_attempt')
countryCode	varchar(3)
countryName	varchar(100)
regionCode	varchar(3)
regionName	varchar(100)
City	varchar(45)
geoipAdded	datetime
asn	int(10) unsigned

Obrázek 10: Tabulka mlp_attribute

2.3 Návrh možných agregací

Jak již bylo zmíněno v předešlé kapitole, agregace dat slouží především k vyjadřování informací v sumární podobě. Z podstaty této definice vyplývá, že by agregace mohly být aplikovány na každý sloupec tabulky mlp_attribute. Nicméně k návrhu agregací by se mělo postupovat metodicky, a nejlépe tak, aby výsledná data měla co největší výpovědní hodnotu. Tímto postupem se mohou jednoduše eliminovat sloupce z tabulky mlp_attribute, které neobsahují patřičné informace.

Cílem této diplomové práce je návrh a implementace agregací, které by potenciálně mohly pomoci při identifikaci utočníků na VoIP systémy, nebo identifikaci specifických nástrojů k tomu určených. Z pohledu síťového provozu jsou nejzajímavější tyto informace:

- Zdroj útoku
- Transportní protokol
- SIP zprávy

Pokud zohledníme tato kritéria, zůstane nám výčet sloupců tabulky mlp_attribute, ze kterých se dá vytěžit nejvíce informací vedoucích k nalezení potencionálního útočníka. Z tohoto výčtu se již dají formovat poměrně zajímavé agregace, jejichž výsledky mohou být základem důkladnější analýzy ke zjištění potencionálních útočníků nebo nástrojů sloužících k útokům na VoIP systémy.

S ohledem na zmíněná kritéria byly zvoleny tyto agregace:

- agregace dat na základě autonomních systémů,
- agregace dat na základě šiřitelů škodlivého provozu,
- agregace dat na základě dynamického útočného okna,

- agregace dat na základě signalizačního protokolu SIP.

Každá z těchto navržených agregací poskytuje jiný náhled na data ze systému Beekeper a může vést k jiným závěrům. Tyto agregace jsou detailně popsány v Kapitole 3.

2.4 Rozšíření stávajícího databázového modelu

Databáze systému Beekeeper se skládá z několika tabulek, jejichž jádrem je tabulak mlp_attribute. Tento systém byl založen pod jinými kritériemi, které se plně neshodují s potřebami této diplomové práce. Databázový model je proto třeba rozšířit o pomocné tabulky.

Vezmeme-li v úvahu první tři agregace a informace uložené v tabulce mlp_attribute, pak je patrné, že informace v ní nejsou dostačující. Je zapotřebí získat data o dostupných autonomních systémech a subnetech, do kterých patří všechny IP adresy, které jsou uložené v databázi. Nejjednodušším způsobem, jakým získat tyto informace, je přímo z pátečních BGP směrovačů. Existují však i veřejné projekty, které tyto data poskytují právě za účelem další analýzy. Jedním z těchto projektů je i tzv. *Routing report*, jenž je volně dostupný z adresy http://thyme.apnic.net/. Tento projekt spadá pod regionální internetový registr APNIC, který poskytuje veškerá data. Data pochází z několika geologicky rozmístěných směrovačů. Výhoda možnosti výběru dat z několika zařízení je v tom, že pokud na některém zařízení chybí data, např. kvůli výpadku, můžeme pohodlně využít dat z jiného zařízení.

Pro ukládání dat o IP adresách a autonomních systémech byly vytvořeny dvě tabulky: *BGP_info* a *IP_info*. Obě tabulky jsou si velmi podobné, nicméně pro splnění svého účelu jsou nezbytné.

Tabulka *BGP_info* slouží k uložení informací o subnetech, namapovaných do korespondujících autonomních systémů ze všech dnů, jenž jsou přístupné v tabulce *mlp_attribute*. Byť existují služby a protokoly, kterými by se dalo získat informace o konkrétním subnetu v reálném čase (např. protokol Whois), není možné uvažovat o jejich použití. [12] Právě protokol Whois má implementován mechanismus, který má zabránit potencionálnímu zneužití služby. Takto je omezen počet dotazů za den, kterými by mohli být získány informace o konkrétní IP adrese nebo subnetu. Omezení se sice mohou lišit podle domén, nicméně to nic nemění na situaci, že nelze provádět tisíce nebo desetitisíce dotazů denně. Z tohoto důvodů je praktičtější udržovat lokální databázi s potřebnými informacemi, které jsou k dispozici kdykoli. Na Obrázku 11 lze vidět detail tabulky *BGP_info* obsahuje následující sloupce:

- **id** - identifikátor konkrétního záznamu,
- **asn** - číslo autonomního systému, bez prefixu AS,
- **subnet** - subnet, patřící do konkrétního autonomního systému,
- **date** - datum, ze kterého byla data pořízena,

- **rtr_geo_info** - informace o geolokaci směrovače, ze kterého data pochází.

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
asn	int(10) unsigned	YES	MUL	NULL	
subnet	varchar(20)	YES		NULL	
date	date	YES	MUL	NULL	
rtr_geo_info	varchar(10)	YES		NULL	

Obrázek 11: Entita BGP_info.

Tabulka *IP_info* obsahuje informace o konkrétních IP adresách z tabulky *mlp_attribute*, které jsou namapovány do korespondujících subnetů a autonomních systémů. Tato tabulka současně slouží jako základ pro několik agregací, jelikož obsahuje klíčové informace, jako jsou právě ASN nebo datum. Na Obrázku 12 je znázorněna tabulka *IP_info*, která obsahuje následující sloupce:

- **id** - identifikátor konkrétního záznamu,
- **asn** - autonomní systém, do kterého patří konkrétní IP adresa,
- **subnet** - subnet, do kterého patři konkrétní IP adresa,
- **ip** - konkrétní IP adresa z tabulky *mlp_attribute*,
- **date** - datum, ze kterého je záznam o IP adrese z tabulky *mlp_attribute*.

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
asn	int(10) unsigned	YES	MUL	NULL	
subnet	varchar(20)	YES		NULL	
ip	varchar(15)	YES		NULL	
date	date	YES	MUL	NULL	

Obrázek 12: Entita IP_info.

3 Implementace, vizualizace a testování agregací

V minulé kapitole byly popsány možné agregace, které by mohly mít vliv na odhalení potenciálních útočníků či identifikaci útočných nástrojů. Tato kapitola se bude věnovat detailnějšímu popisu jednotlivých agregací včetně teoretického rozboru jejich implementací.

Pro implementaci agregačních algoritmů a funkcí, exportujících data pro vizualizaci, byl použit programovací jazyk Python ve verzi 3.7.1. Pro samotné vizualizace byl zvolen skriptovací jazyk JavaScript a jeho knihovna Highcharts, která má velmi rozsáhlou dokumentaci pro tvorbu různých grafů a map. Implementace probíhala ve virtuálním prostředí s operačním systémem Linux Mint 19 XFCE. Vzhledem k tomu, že některé části kódu slouží k práci se soubory, je výhradně doporučeno spouštění celého programu v operačních systémech založených na OS Linux.

3.1 Agregace dat na základě autonomních systémů

První, a zároveň nejdůležitější agregací je agregace dat na základě autonomních systémů. Tato agregace totiž ukládá data do tabulek *bgp_info* a *ip_info*. Jak je již patrné z názvu, algoritmus mapuje jednotlivé IP adresy do jejich korespondujících autonomních systémů. Pro mapování byl využit volně dostupný projekt *Routing report*, jak již bylo uvedeno v kapitole 2.4.

Agregace je rozdělena do několika fází. Každá z těchto fází má svůj specifický úkol a výstup:

- Fáze 1 - kontrola dat
- Fáze 2 - získání dat z *Routing report*
- Fáze 3 - mapování IP do autonomního systému

Všechny fáze řídí funkce *aggregate_to_asns*, která má na svém vstupu čtyři parametry:

- *connection* - objekt sezení s databází
- *cursor* - kurzor sloužící pro správu operací s databází
- *start_date* - počáteční datum zvolené uživatelem
- *end_date* - poslední datum zvolené uživatelem

Parametry se dle potřeby předávájí do dalších funkcí v jednotlivých fázích.

V první fázi probíhá jednoduchá kontrola dat. Pokud data za konkrétní den neexistují v tabulce *bgp_info*, lze předpokládat, že neexistují namapovaná data v tabulce *ip_info*, jelikož data z této tabulky jsou závislá na datech z tabulky *bgp_info*.

Ve druhé fázi probíhá získání a zpracování dat z projektu *Routing report*. Aby mapování bylo co nejpřesnější, je zapotřebí získat data o subnetech v jednotlivých autonomních systémech za

každý den, kdy chceme vidět výsledky agregace. A to z toho důvodu, že vlastníci konkrétních subnetů nebo supernetů se mohou měnit. V této fázi rovněž probíhá práce se získanými datovými soubory, což z velké části zahrnuje úpravu souborů do podoby vhodné k nahrání do databáze. Jakmile jsou soubory patřičně upraveny, může dojít k samotnému nahrání. Ukázka upravených dat z jednoho z datových souborů je na Obrázku 13.

```
20940          2.16.8.0/24
20940          2.16.9.0/24
20485          2.16.10.0/24
20940          2.16.11.0/24
16625          2.16.12.0/23
 1273          2.16.14.0/23
 1273          2.16.16.0/23
34164          2.16.18.0/24
 6762          2.16.19.0/24
16625          2.16.20.0/23
```

Obrázek 13: Ukázka upravených dat ze souboru *data-add-ripe*.

Se soubory lze pracovat lokálně a obejít se bez tabulky *bgp_info*, nicméně takto můžeme k datům přistupovat bez toho, aniž bychom je znovu získávaly. Dalším důvodem je rychlost, protože vyhledávání indexovaných dat v databázi je mnohem rychlejší než na lokálním disku. Získání a zpracování dat probíhá ve funkci *get_bgp_data*. Na Obrázku 14 je ukázka dat z tabulky *bgp_info*.

id	asn	subnet	date	rtr_geo_info
1	13335	1.0.0.0/24	2018-12-18	London
2	56203	1.0.4.0/22	2018-12-18	London
3	56203	1.0.4.0/24	2018-12-18	London
4	56203	1.0.5.0/24	2018-12-18	London
5	56203	1.0.6.0/24	2018-12-18	London

Obrázek 14: Ukázka dat z tabulky *bpg_info*.

Třetí fáze zahrnuje samotné mapování IP adres do autonomních systémů. Tato fáze je rozdělena do dvou funkcí: *select_subnets* a *ip_to_asn*. Funkce *select_subnets* vybírá data na základě prvních tří oktetů IP adresy, kterou chceme mapovat do patřičného autonomního systému, z tabulky *bgp_info* a má tři parametry:

- *cursor* - kurzor sloužící pro správu operací s databází,
- *ip* - ip adresa pro mapování,
- *date* - datum záznamu z tabulky *mlp_attribute*.

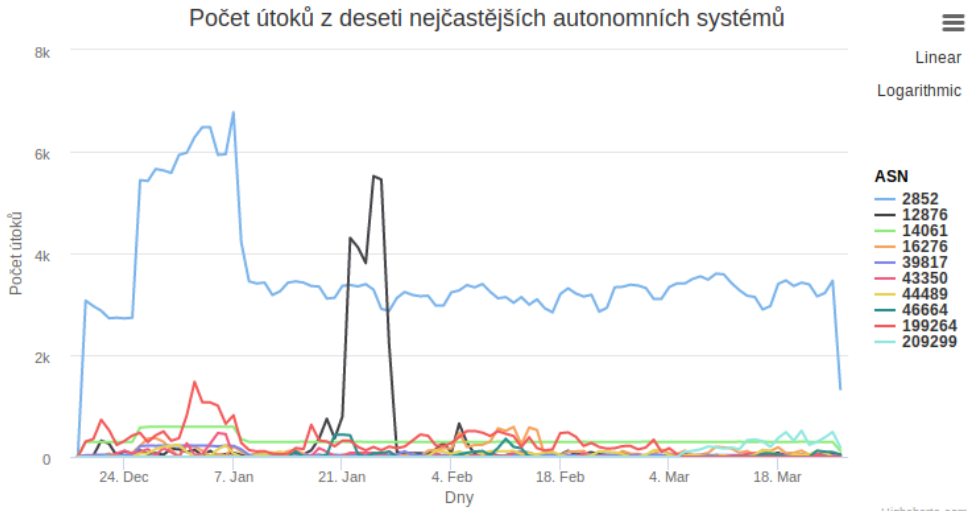
Výsledek výběrového dotazu, spolu s ostatními parametry, je předán funkci *ip_to_asn* pro mapování. Mapování probíhá na základě nejvyšší shody prvních n bitů binárního tvaru IP adresy a konkrétního subnetu z tabulky *bgp_info*, s ohledem na masku sítě. Jakmile je nalezen subnet s největší shodou, jsou všechna relevantní data zapsána do souboru, který se po ukončení algoritmu nahraje do databáze, konkrétně do tabulky *ip_info*.

id	asn	subnet	ip	date
1	2852	146.102.0.0/16	146.102.0.40	2018-12-18
2	2852	147.228.0.0/16	147.228.1.41	2018-12-18
3	2852	147.228.0.0/16	147.228.210.5	2018-12-18
4	2852	147.228.0.0/16	147.228.212.20	2018-12-18
5	2852	147.228.0.0/16	147.228.212.21	2018-12-18

Obrázek 15: Ukázka dat z tabulky *ip_info*.

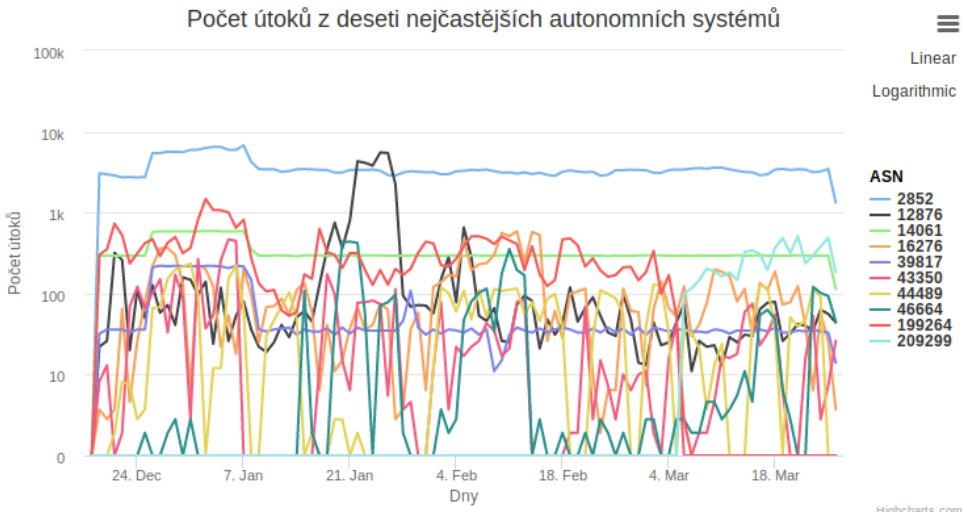
3.1.1 Vizualizace agregace

Díky této agregaci můžeme sledovat, odkud jednotlivé útoky pochází, jak často k nim dochází a celkové vytížení jednotlivých autonomních systémů. Vzhledem k těmto faktorům byly vytvořeny grafy, které zobrazují průběhy nejškodlivějších autonomních systémů v čase. Tyto grafy obsahují celkem dvě měřítka - lineární a logaritmické. Zároveň je zobrazen popisek při posunutí kurzoru na jednotlivé časové úseky, přičemž je možnost výběru časového úseku pro detailnější analýzu. V grafu lze rovněž vypnout křivku jakéhokoli autonomního systému a tím sledovat jen ty, které jsou pro analýzu nejzajímavější. První z těchto grafů je zobrazen na Obrázku 16, na kterém je zobrazen počet útoků z deseti nejškodlivějších autonomních systémů za dané období v lineárním měřítku. Zobrazený graf obsahuje data za období od 18.12.2018 do 26.3.2019.



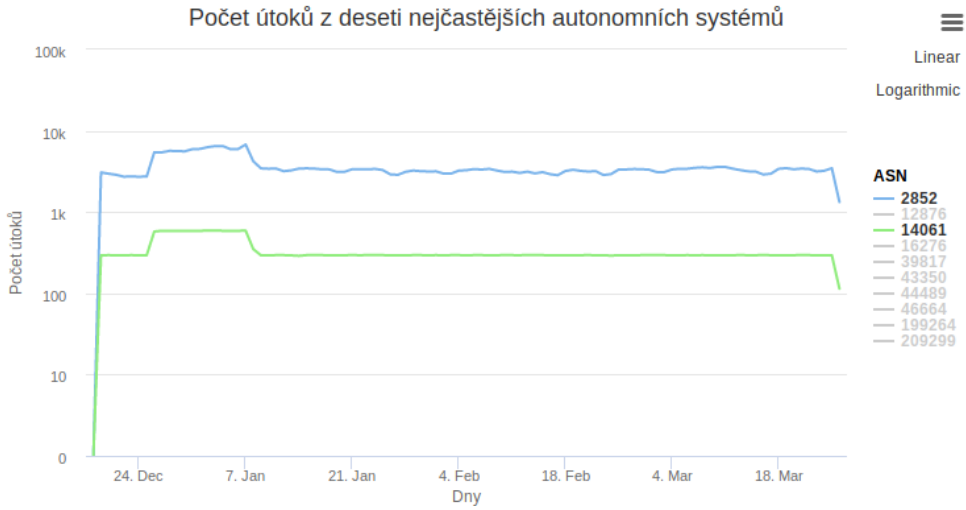
Obrázek 16: Počet útoků z deseti nejčastějších ASN - lineární měřítko.

Z grafu je patrných několik věcí. Nejpatrnější je však autonomní systém AS2852, který počtem útoků vysoce převyšuje ostatní autonomní systémy. Rovněž je u něj vidět prudký nárůst útoků v období od 26.12.2018, které dále pomalu rostou s mírným kolísáním, dokud nedosáhnou svého nejvyššího bodu dne 7.1.2019. V tuto dobu dochází k prudkému poklesu do normálního režimu. V grafu je však patrná ještě jedna špička, a to špička v útocích autonomního systému AS12876. Nicméně, v logaritmickém měřítku stejného grafu lze najít jiné zajímavosti, nebo podobnosti v datech, které by v lineárním měřítku nemusely být odhaleny. Tento graf je zobrazen na Obrázku 17, a na první pohled je velmi nepřehledný, ale skrývají se v něm zajímavé informace.



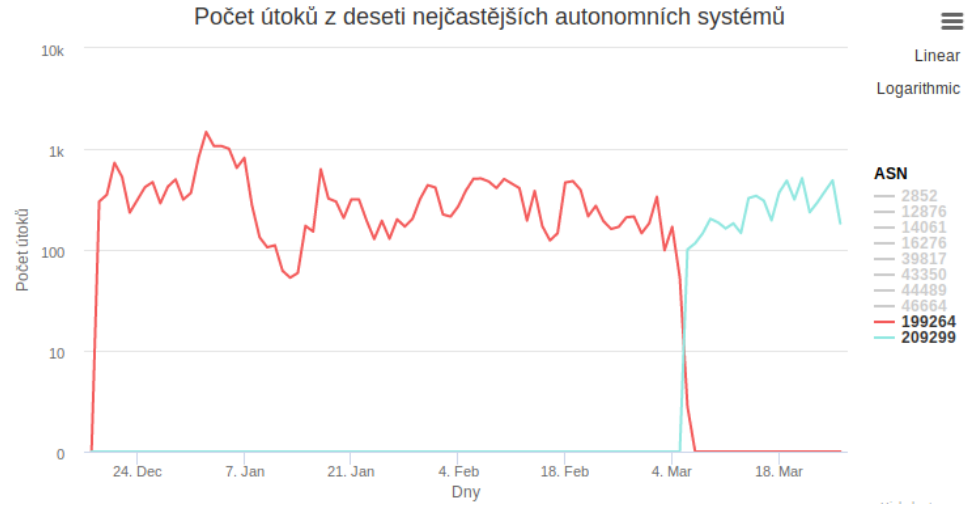
Obrázek 17: Počet útoků z deseti nejčastějších ASN - logaritmické měřítko.

První zajímavou informací je až nápadná podobnost průběhu útoků u autonomních systémů AS2852 a AS14061, kterou si můžeme prohlédnout na Obrázku 18. Dalo by se říci, že průběh útoků těchto autonomních systémů je téměř totožný, což může, ale také nemusí vést k rozdílným závěrům.



Obrázek 18: Podobný průběh útoků u AS2852 a AS14061.

Druhou zajímavou věcí je průběh útoků u autonomních systémů AS199264 a AS209299. Jak je patrné z Obrázku 19, průběh útoků AS199264 poměrně fluktuje. Jakmile dojde k úplnemu ukončení útoků, začnou se útoky projevovat na druhém zmíněném autonomním systému. Zde však podobnost nekončí pouze na úrovni grafu, protože oba autonomní systémy patří Nizozemské firmě *VITOX TELECOM*, která bývá často pod terčem útoků. V období od 1.12.2018 do 1.3.2019 měla dvě IP adresy v seznamu nejčastějších adres, které byly oblíbenou destinací pro útok. Na jedné ze zmíněných adres, konkrétně adresa *185.53.88.46*, se objevily útoky i na systém Beekeeper. Bylo jich celkem 27.

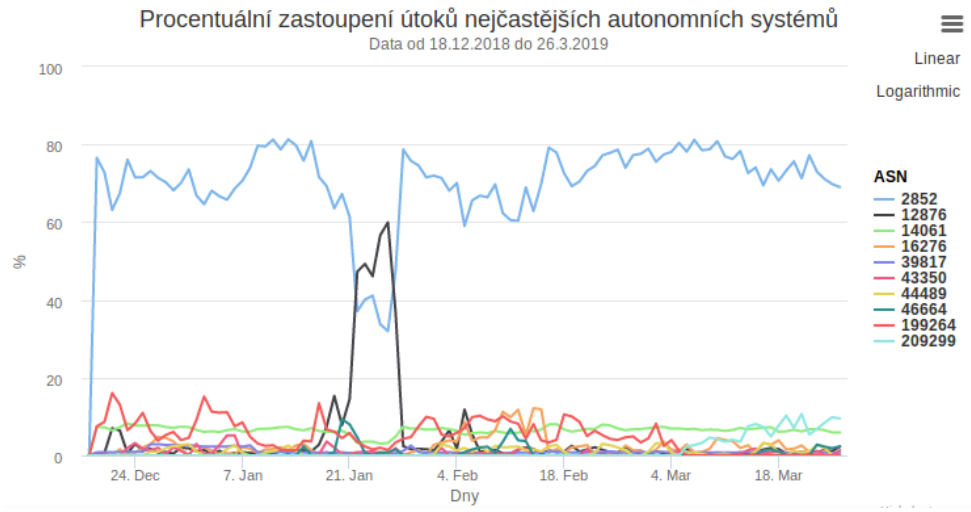


Obrázek 19: Průběh útoků u AS199264 a AS209299.

Posledním grafem zobrazujícím výsledky této agregace je graf procentuálního zastoupení útoků deseti největších autonomních systémů. Jak již šlo vypozorovat z předchozích grafů, nej-

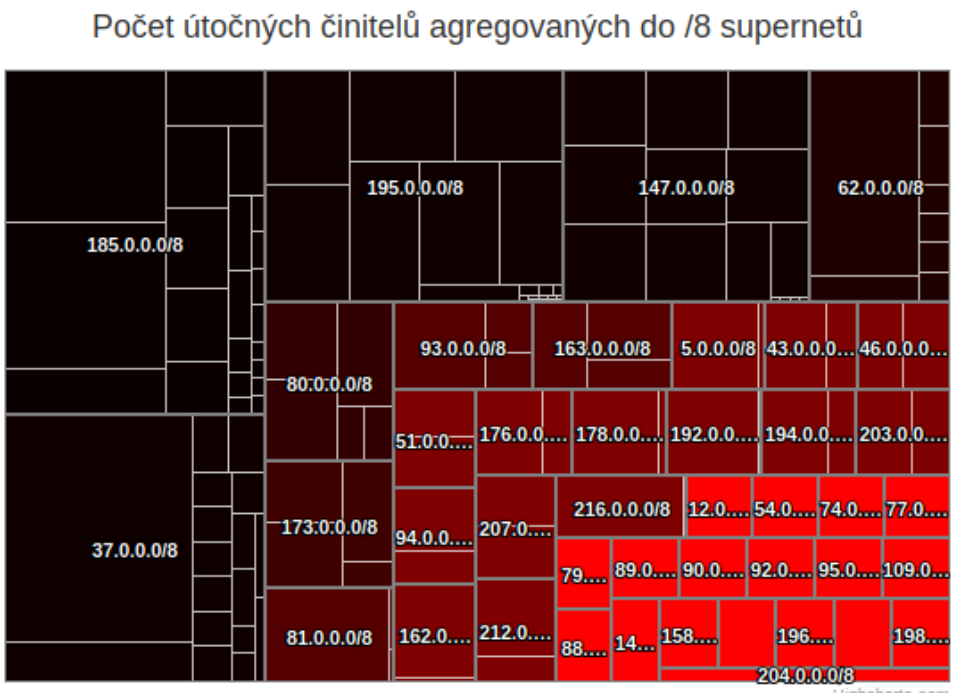
větší podíl na útocích má AS2852, jehož procentuální zastoupení se pohybohuje mezi 60-80 %.

U AS12876 dochází k výkyvu v datech, který ve své špičce dosahuje na téměř 60 % počtu útoků v daném období. Počet útoků u ostatních autonomních systémů fluktuje mezi 0-10 %, kde v ojedinělých případech dochází k nárůstu na 16 %, 15 % a 13.5 % u AS199264 a 12 % u AS16276.



Obrázek 20: Procentuální zastoupení útoků TOP10 ASN.

Tato agregace rovněž umožnila vizualizaci jednotlivých činitelů škodlivého provozu podle masky podsítě proměnlivé délky. Na Obrázku 21 lze vidět jednotlivé činitele agregované do /8 supernetů. Graf obsahuje data za den 1.1.2019. Je z něj patrné, že největším supernetem byl v tento den supernet 185.0.0.0/8, který čítal 22 činitelů. Při zobrazení detailu tohoto supernetu lze vidět jednotlivé činitele s počtem vykonaných útoků. Největším z těchto činitelů byl činitel 185.107.83.89, který vykonal celkem 241 útoků.



Obrázek 21: Počet útočných činitelů agregovaných do /8 supernetů za den 1.1.2019.

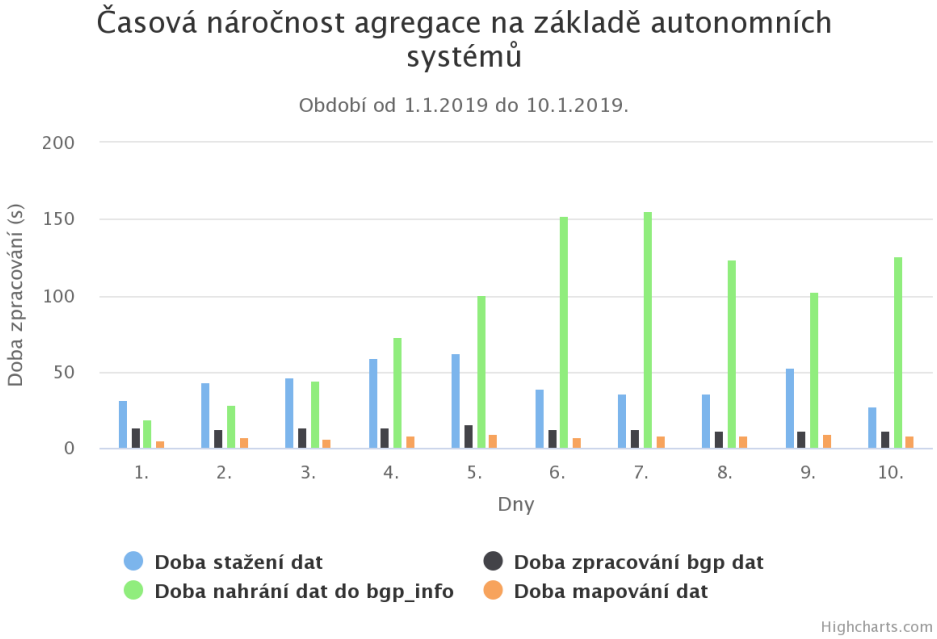
3.1.2 Výkonnostní testování agregace

Tří fázový proces je poměrně časově náročný, jelikož zde dochází k operacím, které mohou významně ovlivnit dobu zpracování agregace. Pro lepší pochopení časové náročnosti agregace byl vytvořen interaktivní graf. V tomto grafu lze sledovat čtyři parametry:

- dobu potřebnou stažení BGP dat,
- dobu potřebnou pro zpracování BGP dat,
- dobu nahrání dat do tabulky bgp_info,
- dobu mapování ip adres do autonomních systémů.

Graf časové náročnosti jednotlivých kroků je zobrazen na Obrázku 22. Z grafu je patrných hned několik věcí. Nejvíce časově náročná je zjevně první fáze celé agregace, kde probíhá získání dat, zpracování a nahrání do databáze. Jelikož se jedná o statisíce BGP záznamů, dá se předpokládat, že nahrání do databáze a stažení souborů bude časově náročnější operace. U stažení souborů hraje ještě roli rychlost a stabilita připojení, která může velmi ovlivnit dobu potřebnou pro stažení souborů. Tyto soubory mají zpravidla velikost okolo 16MB. Počet záznamů v těchto souborech se rovněž liší. Zpravidla se jedná o rozmezí mezi 710000-750000 záznamy. Zde závisí na tom, jaké informace propagují jednotlivé ASN. Z grafu je rovněž patrný nárůst doby potřebné pro nahrání dat do databáze. To je pravděpodobně způsobeno tím, že do určitého počtu dat

se pro zpracování a přepočítávání indexů využívá jiného mechanismu než se začalo využívat 8.1.2019., kdy došlo ke zlomu.



Obrázek 22: Graf časové náročnosti agregace na základě autonomních systémů za období 1.1.2019-10.1.2019.

Naopak nejméně časově náročná operace je samotné mapování IP adres na autonomní systémy. Zde můžou rychlost mapování nejvíce ovlivňovat dva faktory:

- počet adres určených k mapování,
- počet možných subnetů, do kterých adresa může patřit.

První faktor závisí na počtu škodlivých adres, které mají záznam v tabulce *mlp_attribute*. Druhý faktor je pak dám tím, které subnety jednotlivé autonomní systémy propagují do směrovacího protokolu BGP. Algoritmus totiž nejprve hledá shodu v prvních třech oktetech mezi mapovanou IP adresou a záznamy v tabulce *bgp_info*. Pokud shodu nenajde, snaží se o nalezení shody mezi prvními dvěma oktety, a následně pouze v prvním oktetu. Takto se dá významně snížit okruh hledaného subnetu a autonomního systému, do kterého patří. Nicméně ani jeden z těchto faktorů nemůžeme nijak ovlivnit. U zpracování se tedy můžeme pohybovat v řádech sekund u desítek až stovek IP adres nebo řádech minut pro tisíce až desetitisíce.

3.2 Agregace dat na základě šiřitelů škodlivého provozu

Jedná se o jednodušší agregaci, na kterou můžeme pohlížet z více úhlů. Při pohledu opět na Obrázek 10, kde je zobrazena tabulka *mlp_attribute*, tak lze vidět, že pro jednotlivé šiřitele

můžeme agregovat různé sloupce. O šiřiteli škodlivého provozu je třeba uvažovat jako o IP adrese, která má záznam v tabulce *mlp_attribute*. Tudíž jeden záznam v tabulce lze brát jako potencionální škodlivý provoz. Tímto způsobem bychom mohli agregovat např.:

- zdrojové porty,
- počet připojení,
- využití jednotlivých SIP zpráv,
- čas útoku,
- oblast,

a to vše vztaženo na jednu konkrétní IP adresu. Co se však týče informační hodnoty, tak nejvíce zajímavý je počet "hitů"neboli počet útoků z konkrétní IP adresy vztažen na konkrétní časový úsek. Tímto lze pozorovat, jak se v čase mění využití šiřitelů škodlivého provozu, zda se jedná o periodicky opakující se události, nebo o jednorázové pokusy.

3.2.1 Vizualizace agregace

Vzhledem k tomu, že se jedná o poměrně jednoduchou agregaci, nebyl implementován přímý mechanismus pro zpracování. Naproti tomu bylo vytvořeno několik funkcí, které jsou založeny na této jednoduché agregaci a přímo generují výstupní data v podobě *json* souborů pro vizualizaci nebo další statistické zpracování. Mezi tyto funkce patří:

- *data_to_json_attack_source_usage_hist*,
- *data_to_json_attack_source_usage_per_asn*.

Názvy obou funkcí naznačují, k jakému účelu jsou data zpracovávána. První z těchto funkcí slouží k jednoduchému zobrazení počtu záznamů (útoků), zaznamenaných v tabulce *mlp_attribute*, pro jednotlivé šiřitele škodlivého provozu. Tento graf je zobrazen na Obrázku 23, ve kterém lze vidět data za období od 1.1.2019 do 10.1.2019.

Jak je z grafu patrné, většina šiřitelů škodlivého provozu má naprosto zanedbatelný počet záznamů oproti záznamům s nejvyšším počtem útoků.

Obrázek 23: Histogram počtu útoků z jednotlivých šiřitelů.

Na Obrázku 24 je přiblížena část histogramu. V tomto grafu je ukázán pokles počtu útoků z 550 do 22. Z toho lze pozorovat, že adresy již v této části grafu mají malé počty záznamů. Pro srovnání je v tomto grafu ukázán záznam s informacemi o šiřiteli *207.180.242.195*, jehož počet záznamů v tabulce *mlp_attribute* čítá pouhých 87 záznamů, za již zmíněné období. Oproti tomu šiřitel *195.178.192.243*, jenž je největším šiřitelem škodlivého provozu za toto období, má počet záznamů 3377.

IP adres, jenž mají počet záznamů větších než 3000, je pouhých 14 z celkového počtu 542 adres. Nejzajímavější na této informaci je však to, že všechny tyto IP adresy pocházejí ze stejného autonomního systému, a to AS2852. Některé z těchto adres patří i do stejného subnetu.

Následně je v grafu patrný prudký pokles v počtech záznamů, kde se jednotlivé autonomní systémy více či méně mění. Je tedy na uvážení, jaký počet záznamů, je ještě považován za legitimní provoz a kde je hranice pro detekci škodlivosti.

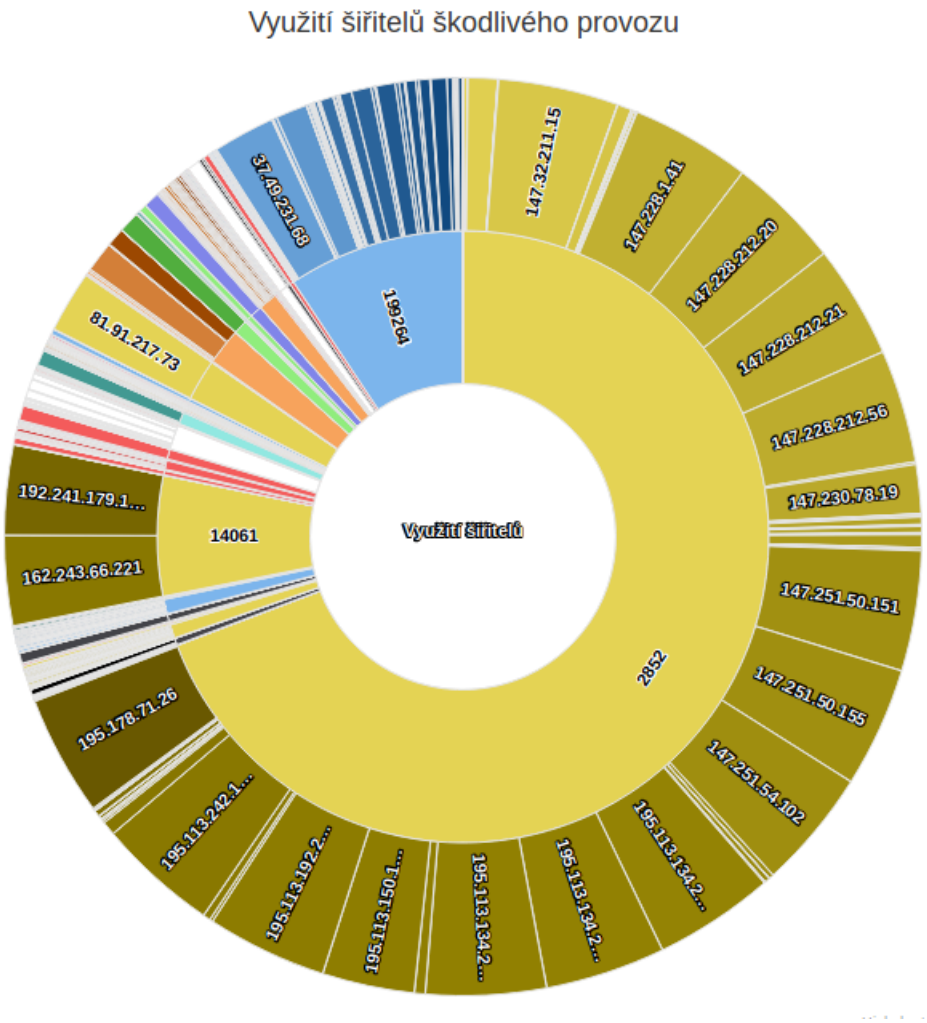
Obrázek 24: Výběr oblasti z histogramu počtu útoků z jednotlivých šiřitelů.

Druhá z uvedených funkcí je již mírně komplikovanější. Nesleduje se zde pouhé využití jednotlivých šiřitelů na úrovni IP adres, ale díky agregace popsané v předchozí kapitole lze sledovat, ze kterých autonomních systémů se škodlivý provoz šíří, a který autonomní systém má největší podíl na celkovém škodlivém provozu. Výsledkem této vizualizace je paprskový graf, který je zobrazen na Obrázku 25.

Graf má celkem tři úrovně, které nesou následující informace:

- Vnitřní část (jádro) grafu - obsahuje informace o celkovém počtu IP adres použitých pro škodlivý provoz, celkový počet autonomních systémů, do kterých patři jednotlivé IP adresy a celkový počet útoků.
- Vnitřní prstenec - obsahuje informaci, do kterého autonomního systému patří IP adresy z vnějšího prstence, celkový počet útoků z toho autonomního systému a celkový počet IP adres, jenž se podílelo na útocích.
- Vnější prstenec - obsahuje jednotlivé IP adresy, z nichž byly vedeny útoky, a informace s nimi spojené, jake je subnet a počet útoků.

Takto můžeme pohodlně sledovat největší šiřitele škodlivého provozu. Z grafu je patrné, že absolutně největším šiřitelem, na úrovni autonomních systémů, je **AS2852**, za již výše zmíněné časové období.

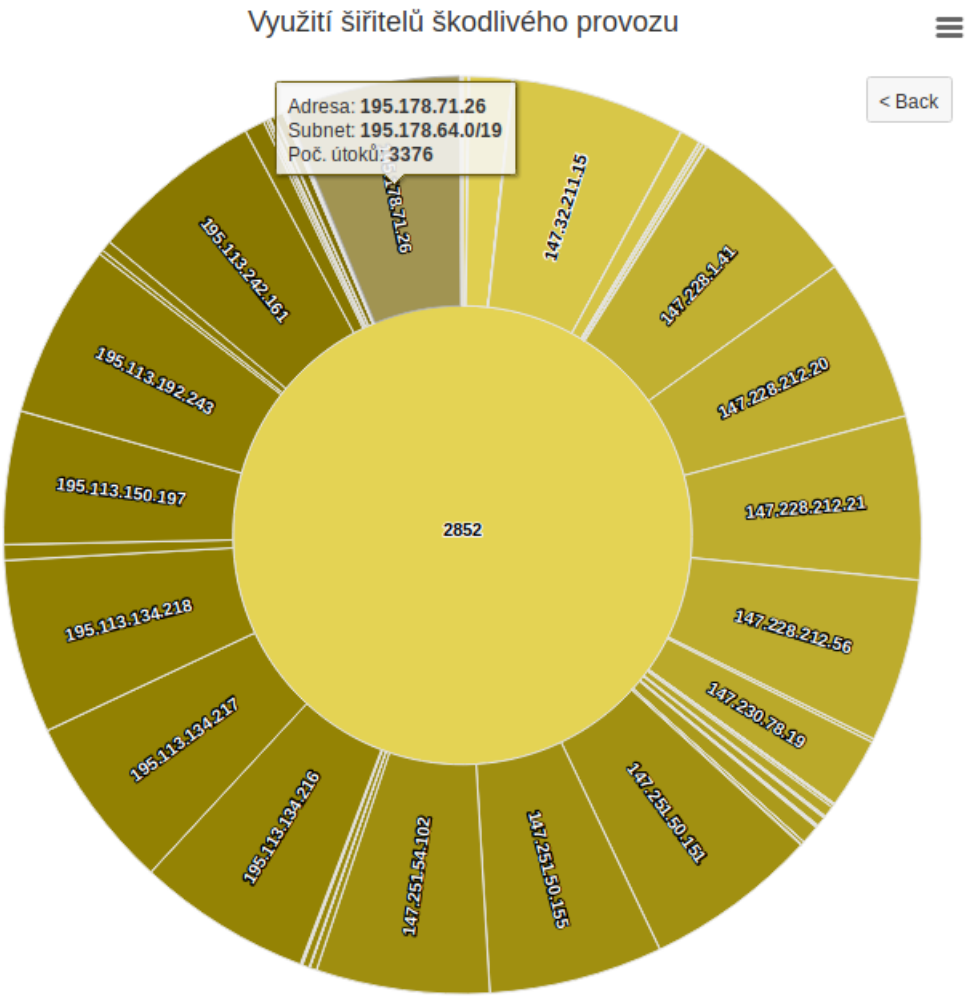


Obrázek 25: Paprskový graf využití šiřitelů škodlivého provozu.

Z grafu jsou patrné další autonomní systémy, které měly rovněž větší zastoupení v počtu útoků. Jedná se o autonomní systémy AS14061 a AS199264. Pro AS14061 jsou patrné dvě IP adresy, ze kterých jde 99.78% škodlivého provozu. Napopak pro AS199264 je patrné, že škodlivý provoz je mnohem více rozložen. Je proto dobré, zaměřit se na konkrétní autonomní systém, či konkrétní IP adresy, a sledovat jakým způsobem se škodlivý provoz mění v čase. Takto můžeme vysledovat, zdali se jedná o anomálie, nebo ne.

Po kliknutí na autonomní systém se zobrazí dvou-úrovňový graf, jehož středem je číslo autonomního systému a vnější prstenec tvoří IP adresy, patřící do tohoto konkrétního autonomního systému. Tímto lze vybrat konkrétní autonomní systém a zobrazit jej pro další analýzu. Na Obrázku 26 lze vidět tento náhled na AS2852. Zde je nejvíce patrných 14 IP adres, které mají počet útoků větší než 3000. Dále pak jednu adresu s počtem útoků 2587 a jednu adresu s počtem útoků 1378. Ostatní adresy již mají téměř zanedbatelně malý počet útoků vzhledem k výše uvedeným adresám. Pro AS2852 bylo za období od 1.1.2019 do 10.1.2019 zaznamenáno celkem

55080 útoků z 87 IP adres. Z toho 46600 útoků pocházelo právě z již zmíněných 14-ti adres. To celkově dělá 84.6 % provozu za toto období.



Obrázek 26: Využití šiřitelů pro AS2852.

Jak již bylo zmíněno výše, jedná se o výpočetně jednodušší úlohy, tudíž nejsou nějak zvlášť časově náročné. Vzhledem k tomu, že veškeré výběrové operace probíhají nad indexovanými sloupci v tabulkách databáze, odpovídá tomu doba zpracování.

3.2.2 Výkonnostní testování agregace

Pro exportování dat tak, aby byly vhodné pro vizualizaci v podobě histogramu uvedeného na obrázku 23, byla potřebná doba pro provedení funkce *1.640* sekund. Stejným způsobem byla měřena doba potřebná pro export dat u druhé funkce, zde však potřebná doba pro výpočet byla *20.962* sekund. Kde data za výše uvedené období čítala 101 autonomních systémů, 533 IP adres a 79503 útoků. Při změně období na 1.1.2019 až 20.1.2019 byla naměřena doba potřebná pro zpracování u první funkce *1.963* sekund, a u druhé funkce *22.872* sekund. Zde se jednalo o 134

ASN, 766 IP adres a 123911 záznamů o útocích. Z uvedených dat lze vidět, že časová náročnost není dáná počtem záznamů v databázi, ale spíše počtem operací, které se nad ní vykonávají.

3.3 Agregace dat na základě dynamického útočného okna

Jedním z významných faktorů při analýze útoků na VoIP infrastrukturu je čas. Na základě časových údajů lze určit kdy útok začal a kdy skončil. Lze rovněž určit, zdali se škodlivý provoz periodicky opakuje, nebo jestli je časově proměnlivý. Všechny tyto faktory mohou vést k úspěšné identifikaci útoku. Naopak, v časovém úseku hlavní provozní hodiny může být identifikace potencionálního útoku výrazně složitější, jelikož se škodlivý provoz může schovat za legitimní.

Při uvážení těchto faktorů byla implementována agregace na základě dynamicky se měnícího útočného okna. Uvažujeme, že jeden řádek v tabulce *mlp_attribute* je jednotka škodlivého provozu, jeden útok. Dynamické okno má vždy pevně daný časový úsek, ve kterém se očekává, že dojde k dalšímu škodlivému provozu. Pokud k tomuto útoku dojde, okno se posune o tento předem definovaný časový úsek. Takto jsou získány všechny útočné toky za uživatelem definované období. Příklad útočných toků s nastaveným útočným oknem na deset minut je uveden na Obrázku 27.

IP adresa	Čas záznamu	Začátek okna	Konec okna
10.1.0.1	10:00	10:00	10:10
10.1.0.2	10:03	10:00	10:13
10.1.0.5	10:05	10:00	10:15
10.1.0.2	10:06	10:00	10:16
10.1.0.5	10:20	10:20	10:30
10.1.0.9	10:28	10:20	10:38
10.1.0.1	10:35	10:20	10:45
10.1.0.8	11:02	11:02	11:12
10.1.0.7	11:15	11:15	11:25

Obrázek 27: Příklad útočnách toků.

Získání útočných toků probíhá ve funkci *aggregate_attack_streams* a tato funkce má několik parametrů:

- *cursor* - kurzor sloužící pro správu operací s databází
- *start_date* - počáteční datum zvolené uživatelem
- *end_date* - poslední datum zvolené uživatelem
- *window_length* - délka útočného okna

Parametry *start_date*, *end_date* a *window_length* lze ovlivňovat výsledek algoritmu. Nejvýznamější parametr je však *window_length*, jehož nastavením se ovlivňuje velilkost dynamického útočného okna. Pokud se nastaví velikost okna na příliš nízkou hodnotu, nebo naopak na příliš vysokou, může výsledek agregace poskytovat nerelevantní informace.

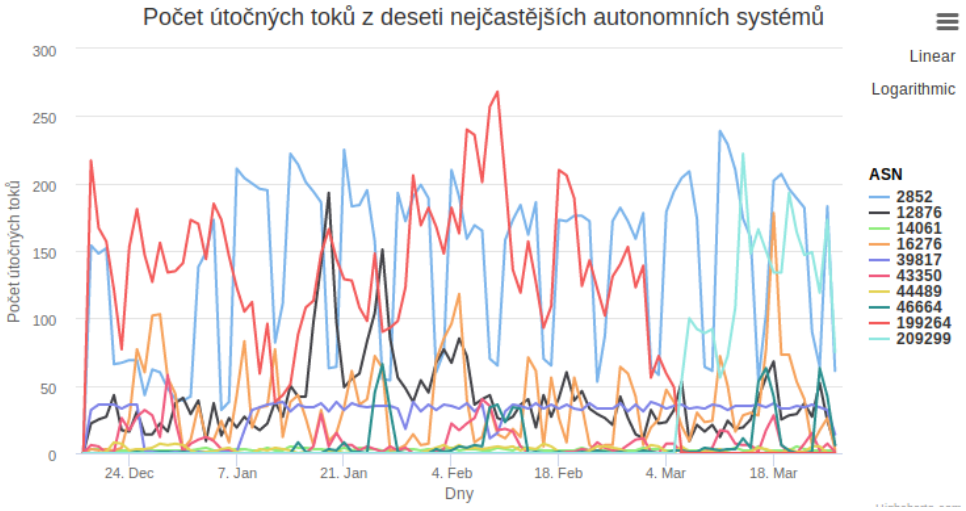
Výstupem této agregace je datová struktura typu *nested list*, kde jednotlivé listy reprezentují jednotlivé útočné toky. Tato datová struktura je dál využívána v několika vizualizačních funkcích.

3.3.1 Vizualizace agregace

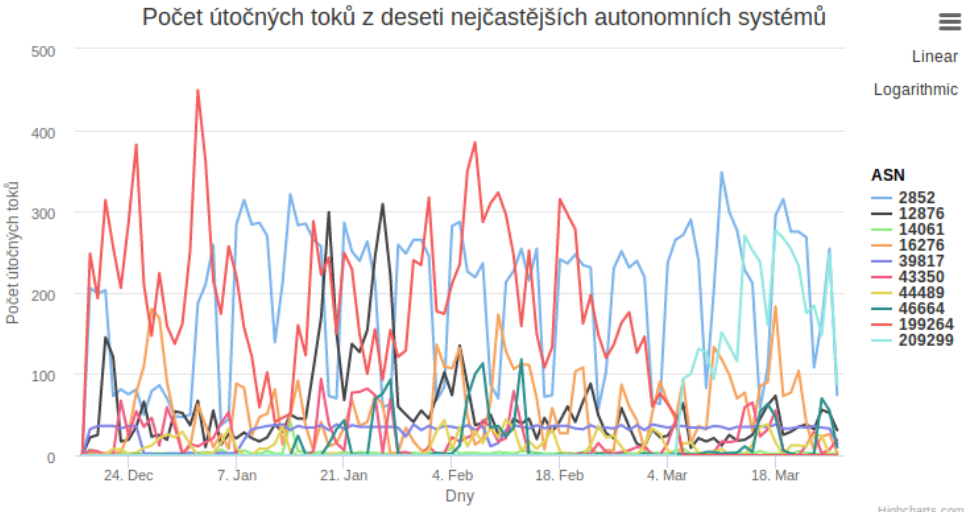
U této agregace máme celkem dvě vizualizační funkce:

- *data_to_json_attack_streams_per_asn*
- *data_to_json_attack_streams_per_asn_tree_map*

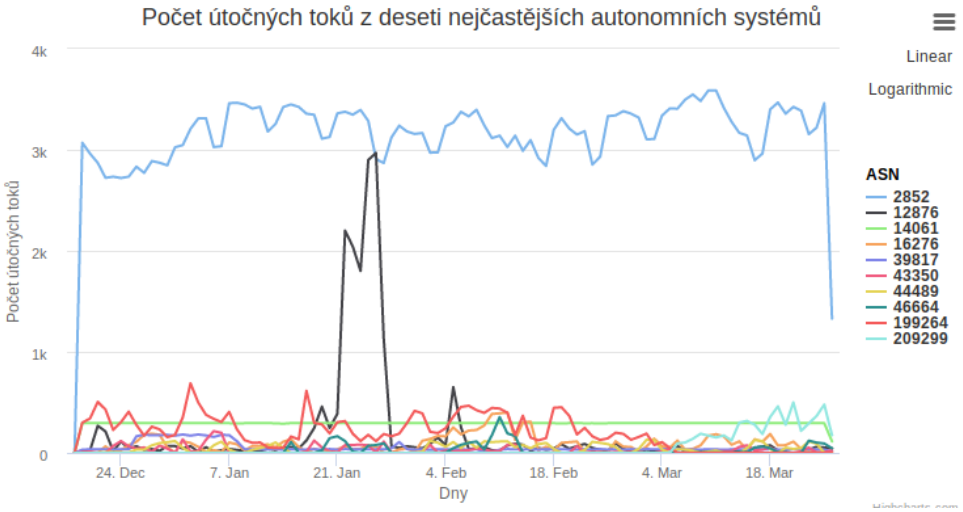
První z těchto funkcí má výstup velmi podobný jako u předchozí agregace. Rozdíl je však v tom, že není sledován počet jednotlivých útoků, ale počet útočných toků. Výsledná data tedy ještě závisí na nastavené délce útočného okna, jak bylo popsáno výše.



Obrázek 28: Počet útočných toků z TOP 10 ASN s délkou okna 20 minut.



Obrázek 29: Počet útočných toků z TOP 10 ASN s délkou okna 10 minut.



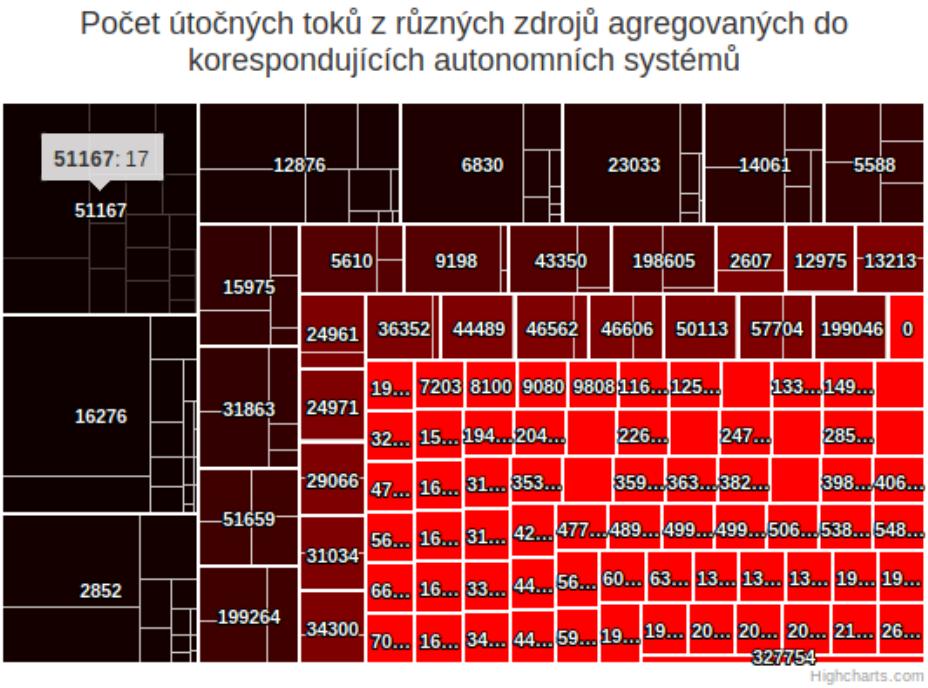
Obrázek 30: Počet útočných toků z TOP 10 ASN s délkou okna 5 minut.

Výše uvedené grafy na Obrázcích 28, 29 a 30 zobrazují průběh počtu útočných toků z deseti nejškodlivějších autonomních systémů za období od 18.12.2018 do 26.3.2019. Velikost dynamického útočného okna pro tyto grafy jsou 20, 10 a 5 minut a lze na nich pozorovat rozdíly v nastavení útočného okna. Graf na Obrázku 30, který má nastavenou velikost útočného okna na pět minut, se už začíná podobat grafu z Obrázku 16. Jak již bylo zmíněno, je třeba dát si pozor na nastavení velikosti útočného okna, jelikož příliš nízká hodnota by mohla mít za následek vykreslení totožného grafu, jako je graf z Obrázku 16. V grafech můžeme pozorovat fluktuaci počtu útočných toků v čase u vybraných autonomních systémů. U některých autonomních systémů, jako jeAS2852, lze vidět jistou konzistenci v počtu útočných toků.

Druhá z uvedených funkcí generuje data pro fundamentálně odlišný typ grafu. Jedná se o tzv. stromovou mapu, která má dvě úrovně: vnější a vnitřní. Vnější stromová mapa zobrazuje jednotlivé autonomní systémy, jenž jsou seřazeny podle počtu subnetů, ze kterých se šíří škodlivý provoz. Fakticky tato stromová struktura obsahuje tři agregace:

- agregaci počtu útoků do útočných toků,
- agregaci útočných toků do korespondujících subnetů,
- agregace subnetů do jejich autonomních systémů.

Na Obrázku 31 je zobrazena stromová mapa s náhledem na AS51167, který obsahuje nejvyšší počet škodlivých subnetů. Největší uvedený autonomní systém, AS51167, nepatří co do počtu útoků, nebo útočných toků, ani do TOP 10 autonomních systémů. Takto se dá na šiřitele nahlížet z jiného pohledu.



Obrázek 31: Vnější úroveň grafu agregujícího útočné toky do subnetů a autonomních systémů.

Na Obrázku 32 ve vnější části grafu lze vidět, že největší počet útočných toků pochází ze subnetu *207.180.242.0/23*. Těchto útočných toků je "pouze"157. U druhého nejškodlivějšího subnetu, tohoto AS, je počet útočných toků třetinový.

Obrázek 32: Vnitřní úroveň grafu z Obrázku 31.

Toto lze odhadnout už podle velikosti jednotlivých bloků. Lze se tedy domnívat, že jednotlivé útočné toky neskrývají velký počet útoků, a proto se tento autonomní systém nedostal do grafů, u kterých byla metrika dána jiným parametrem. Takto se lze podívat na AS2852 jako na nejškodlivější, coby do počtu útoků a útočných toků. Zde je supernet *195.113.0.0/16*, do kterého je agregovaných 803 útočných toků. I když AS2852 značně převyšuje AS51167 v počtu útočných toků a samostatných útoků, je statisticky méně škodlivý v počtu využití jednotlivých subnetů.

3.3.2 Výkonnostní testování agregace

Testována byla pouze funkce *aggregate_attack_streams*, jelikož se jedná o jedinou agregační funkci této agregace. Ostatní funkce slouží už k vizualizaci agregace, a proto nebyly do testování zahrnuty.

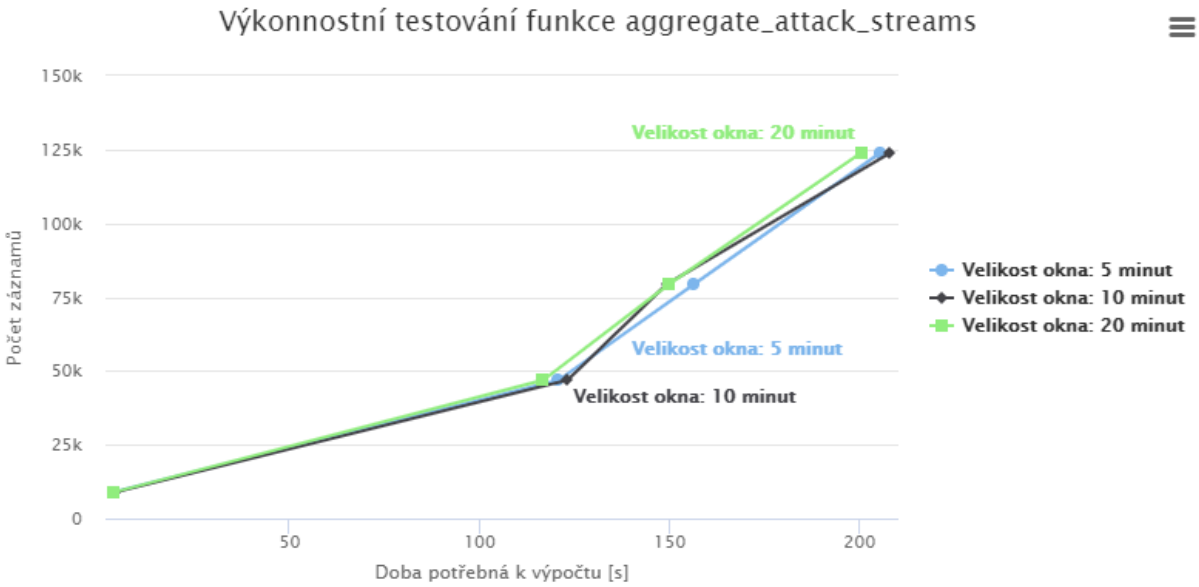
Agregace byla testována v několika časových intervalech, s různou velikostí útočného okna. Časové intervaly byly 1 den, 5 dnů, 10 dnů a 20 dnů. Útočné okno bylo voleno 5 minut, 10 minut a 20 minut. Počtem záznamů se rozumí počet zpracovaných záznamů z tabulky *mlp_attribute*.

Počet dnů	Velikost okna	Počet záznamů	Doba zpracování [s]
1	5	8942	3.53
5	5	46930	120.83
10	5	79530	156.58
20	5	123911	205.54
1	10	8942	4.30
5	10	46930	123.13
10	10	79530	149.39
20	10	123911	208.19
1	20	8942	3.80
5	20	46930	116.78
10	20	79530	150.00
20	20	123911	200.51

Tabulka 3: Data výkonnostního testování funkce *aggregate_attack_streams*.

Výsledky testování agregace lze vidět v Tabulce 3. Na základě výsledků lze bezpečně tvrdit, že časově nejnáročnější je agregace s velikostí útočného okna 10 minut. Dále si z tabulky, a grafu zobrazeného na Obrázku 33, lze povšimnout poměrně velkého nárůstu v době zpracování dat pro jeden a pět dnů. Toto je způsobeno především objemem dat, které musí algoritmus zpracovat pro konkrétní agregaci do útočného okna. Pro jeden den se pohybuje počet záznamů pro adresy s největším počtem útoků v řádech stovek záznamů, ale pro více dnů se lze pohybovat v řádech tisíců. Například pro adresu 147.228.1.41 bylo pro zpracování útočného toku dne 1.1.2019 pouze 392 záznamů, pro pět dní počet záznamů stoupl na 1961, pro deset dní byl počet záznamů 3364

a pro dvacet dní už byl počet záznamů 5325, pouze pro jednu adresu, z jednoho autonomního systému.



Obrázek 33: Graf výkonnostního testování funkce *aggregate_attack_streams*.

3.4 Agregace dat na základě signalizačního protokolu SIP

Předchozí agregace se zaměřují převážně na zdroj veškerého zachyceného provozu, jímž je IP adresa. V této agregaci se nevyužívá zdroj samotné zprávy, nýbrž samotný signalizační protokol SIP. Konkrétně jsou středem zájmu zprávy (žádosti) tohoto protokolu. Základní typy zpráv (žádostí) protokolu SIP byly popsány v kapitole 1.1.3.1. Výčet všech SIP žádostí, nebo také metod, je delší než základních šest typů. Nicméně ze všech metod je nejzajímavějších právě těchto základních šest a metoda *SUBSCRIBE*, která je rovněž uvedena v tabulce *mlp_attribute*.

Agregace je rozdělena do dvou krátých funkcí:

- *aggregate_sip_messages*
- *return_sip_message_count*.

Hlavní funkce *aggregate_sip_messages* slouží k inicializaci proměnných, vytvoření dotazu na databázi, uložení dat do struktury a vytvoření výstupního *json* souboru s agregovanými daty. Tato funkce má definované čtyři parametry:

- *cursor* - kurzor sloužící pro správu operací s databází
- *start_date* - počáteční datum zvolené uživatelem
- *end_date* - poslední datum zvolené uživatelem

- *sip_messages* - výčet SIP zpráv pro agregaci

Parametry *start_date* a *end_date* jsou klíčové parametry, které udávají začátek a konec agregace, a velmi ovlivňují dobu potřebnou pro výpočet. Uživatel může takto ovlivnit, zdali chce vidět agregovaná data za den, týden nebo rok. Uživatel může rovněž ovlivňovat, které SIP metody jej zajímají nejvíc pomocí zaškrtávacích políček ve webové aplikaci, jejichž výstup se projeví v parametru *sip_messages*. Funkce obsahuje jeden cyklus, který slouží pro výběr konkrétní SIP metody. Na základě této metody je složet SQL dotaz, který je dále předán jako parametr do podpůrné funkce *return_sip_message_count*.

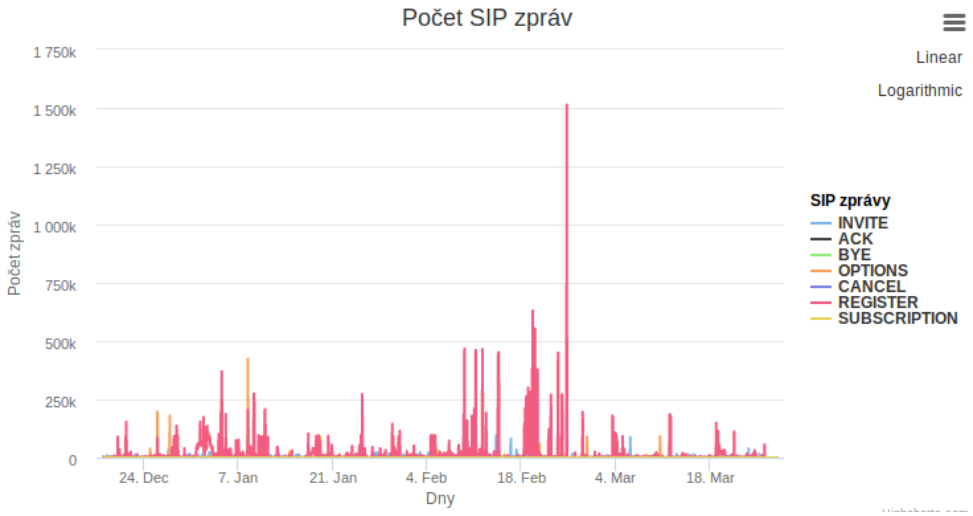
Druhá, nebo také podpůrná, funkce *return_sip_message_count* je volána v každém běhu cyklu a má definován ještě parametr *query*, ve kterém se předává SQL dotaz pro výběr dat z databáze.

Samotná funkce vybírá data na základě uživatelské vstupu (*start_date*, *end_date*). Agregovaná hodnota se vypočte za každou hodinu jednotlivých dnů. Hodnoty jsou spolu následně uloženy do datové struktury *list*, která slouží jako návratová hodnota této funkce. V hlavní funkci jsou data následně uložena do datové struktury typu slovník, ve které je jasně definováno, ke které SIP metodě patří.

Výstupem této agregace je soubor typu *json*, ve kterém jsou data uložena ve tvaru *klíč: hodnota*. Takto zpracovaná data jsou vhodná pro vizualizaci toku SIP zpráv.

3.4.1 Vizualizace agregace

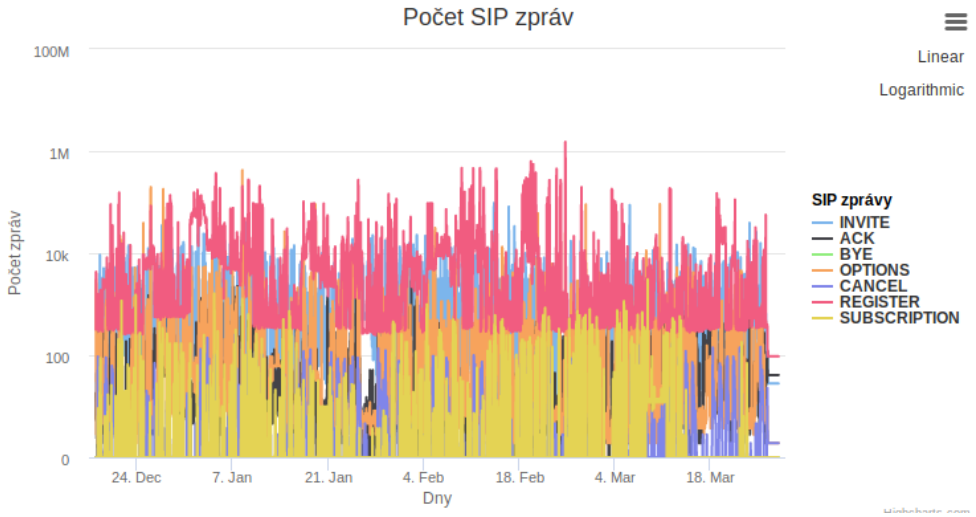
Pro vizualizaci agregace byly vytvořeny celkem dva interaktivní grafy. První z těchto grafů zobrazuje počet SIP zpráv (žádostí) v uživatelem zadaném časovém intervalu. Uživatel má rovněž možnost výběru všech možných žádostí, které se objevily v tabulce *mlp_attribute*. Takto lze sledovat průběh jednotlivých žádostí ve vybraném období. Granularita v tomto grafu byla pro detailnější analýzu nastavena na hodinu. Díky tomu lze využít většího přiblížení a sledovat tak agregovaná data v menších časových intervalech.



Obrázek 34: Vizualizace počtu SIP zpráv.

Na Obrázku 34 je zobrazen první z uvedených grafů. Jak je z grafu patrné, počet určitých SIP žádostí vysoce převyšuje ostatní. Nejvyšší špičky dosahuje žádost *register*, která v ojedinělém nejvyšším bodě dosahuje 1 514 108 žádostí. Tato špička vznikla dne 24.2.2019 mezi 17:00 a 18:00. Vzhledem k tomu, že se jedná o masivní počet žádostí za hodinu, je podezření na útok typu *register flood* velmi vysoké. Tento útok se dá klasifikovat jako DoS/DDoS, jelikož jeho cílem je vyčerpání prostředků SIP serveru. Druhou zajímavou SIP metodou, která dosahuje poměrně velké špičky, je metoda *options*. Ta dosahuje ojedinělé špičky dne 8.1.2019 v 8 hodin, kde dosahuje 422 824 záznamů. Tato metoda slouží k získání informací o daném UA nebo serveru. Pokud je tedy server zaplaven dostatečným počtem požadavků typu *options*, je velmi pravděpodobné, že jej přetížíme. Lze se domnívat, že se jedná o další zaplavovací útok. Ostatní SIP metody sice mají vlastní, někdy ojedinělé, špičky v provoze, nicméně nedosahují takové velikosti, jako již výše zmíněné špičky u metod *register* a *options*.

Na Obrázku 35 je zobrazen totožný graf v logaritmickém měřítku.

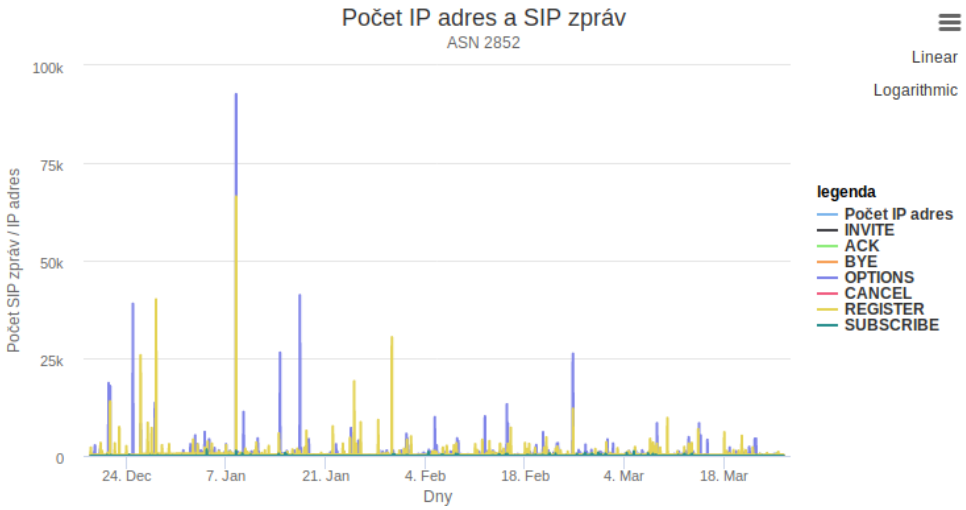


Obrázek 35: Vizualizace počtu SIP zpráv - logaritmické měřítko.

V uvedeném časovém intervalu a všemi metodami je graf nečitelný. Je proto dobré si jednotlivé metody zobrazit zvlášť nebo využít přiblížení pro zkoumání dat v menším časovém intervalu. U metody typu *cancel* lze z tohoto grafu vypozorovat jistá periodicita příchodu žádostí. U ostatních SIP metod lze vidět, že data více či méně fluktují v určitém rozmezí.

Druhým z uvedených grafů vizualizace agregace na základě SIP zpráv, je graf, ve kterém můžeme sledovat *útočný scénář* konkrétního autonomního systému. Probíhají zde dvě agregace. Nejprve se data agregují do uživatelem zvoleného autonomního systému. Následně probíhá agregace dat podle jednotlivých SIP metod, která je však vztažena pouze na konkrétní autonomní systém. Graf útočného scénáře pro AS2852 je zobrazen na Obrázku 36. Původní funkce pro agregaci SIP zpráv *aggregate_sip_messages* nebyla vhodná pro použití v této vizualizaci, a proto byla vytvořená druhá agregační funkce *aggregate_asn_sip_scenario*, která mimo parametry *cursor*, *start_date* a *end_date* obsahuje ještě parametr *asn*, který nese informaci o autonomním systému. Tímto lze docílit znázornění průběhu útočného scénáře.

Tento graf obsahuje informace o počtu jednotlivých SIP metod a zároveň o počtu IP adres, které v daném časovém intervalu šířily škodlivý provoz. V grafu lze vidět, jak se určité SIP metody překrývají, kde by mohla vzniknout logická posloupnost v průběhu útoku. V datech v období od 8.1.2019 do 9.1.2019 lze vidět největší špičku u metody *options*, která se téměř kryje se špičkou metody *register*. Následují špičky u metod *subsription*, *ACK*, *invite* a *cancel*. Tento scénář se opakuje ještě pro několik dalších časových úseků. Z těchto dat lze odvodit určitý scénář útoku, který začíná zaplavováním serveru zprávami typu *options*. Tam, kde útočník úspěšně získá informace o serveru začne být server zaplavován zprávami typu *subscription*. Tímto způsobem útok dále pokračuje a vyplýtváním zdrojů serveru nebo zdrojů útočníka.



Obrázek 36: Útočný scénář AS2852 v období od 18.12.2018 do 26.3.2019.

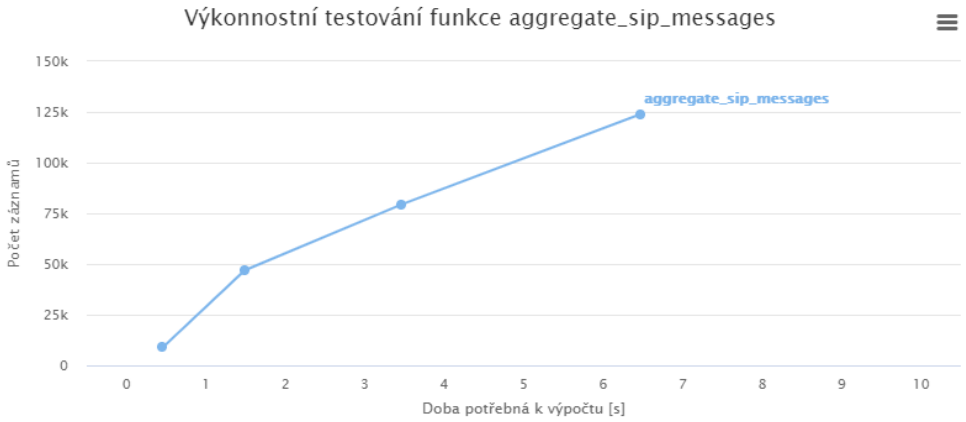
Graf lze přepnout do logaritmického měřítka a sledovat tak podobnosti v datech v jiné škále. V tomto měřítku lze vidět, že SIP metody typu *invite*, *ack* a *cancel* fluktují, spolu s měnícím se počtem IP adres, ve stejných intervalech. Je rovněž vidět nárůst počtu IP adres v období, kdy dochází k největší špičce v provoze. Poslední zajímavou informací je velmi obdobný průběh měnícího se počtu IP adres, a průběh metody register.

3.4.2 Výkonnostní testování agregace

Testování výkonnosti agregace probíhalo zvlášť pro obě agregační funkce: *aggregate_sip_messages* a *aggregate_asn_sip_scenario*. Obě funkce však byly testovány stejným způsobem. Byla testována doba potřebná pro agregaci dat v různých časových intervalech, ty byly 1 den, 5 dnů, 10 dnů a 20 dnů. Pro obě funkce byly v testování zahrnuty všechny SIP metody, a pro druhou zmíněnou funkci byl použit AS2852. Výsledky testů jsou zobrazeny v Tabulkách 4, 5 a na Obrázcích 37, 38. Počtem záznamů se rozumí počet zpracovaných záznamů z tabulky *mlp_attribute*. V Tabulce 4 a na Obrázcích 37 lze vidět jak se mění doba potřebná pro výpočet funkce *aggregate_sip_messages*. Z grafu je patrné, že doba potřebná pro výpočet agregace neroste lineárně, ale logaritmicky.

Počet dnů	Počet záznamů	Doba zpracování [s]
1	8942	0.40
5	46930	1.49
10	79530	3.47
20	123911	6.38

Tabulka 4: Data výkonnostního testování funkce *aggregate_sip_messages*.

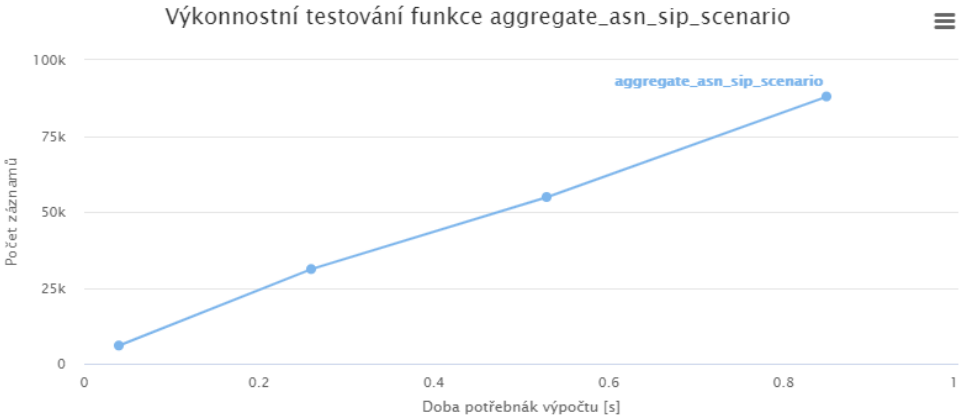


Obrázek 37: Výkonnostní testování funkce *aggregate_sip_messages*.

Výsledky testování pro druhou funkci *aggregate_asn_sip_scenario* jsou zobrazeny V tabulce 5 a grafu 38. Z dat a grafu je patrné, že doba potřebná pro zpracování této agregace má mnohem více lineární průběh, než u první funkce. To je pravděpodobně způsobeno tím, že pro zpracování vybíráme data pouze z jednoho autonomního systému, a tudíž není potřeba provádět více operací v algoritmu.

Počet dnů	Počet záznamů	Doba zpracování [s]
1	5980	0.04
5	31159	0.26
10	54876	0.53
20	87979	0.85

Tabulka 5: Tabulka dat výkonnostního testování funkce *aggregate_asn_sip_scenario*.



Obrázek 38: Výkonnostní testování funkce *aggregate_asn_sip_scenario*.

4 Webové uživatelské rozhraní.

Pro práci s agregacemi a jejich vizualizacemi bylo vytvořeno jednoducé uživatelské rozhraní v podobě webové aplikace. V tomto rozhraní může uživatel volit celkem ze sedmi formulářů. Tyto formuláře slouží jako uživatelský vstup pro zpracování agregací a následné vykreslení grafů.

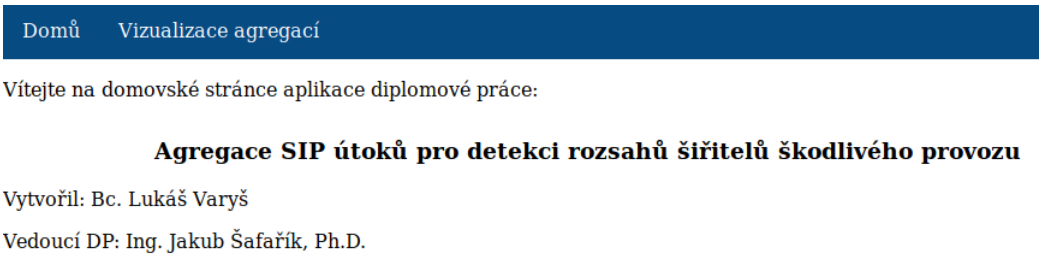
Pro jednoduché nasazení aplikace byl použit webový mikroframework Flask. Flask využívá tzv. "směrování", které slouží pro mapování konkrétního URL s funkcí, která bude vykonávána. Takto je vytvořeno velmi jednoduchého a efektivního propojení mezi aplikačním frontendem a backendem.

Pro spuštění celé aplikace z příkazového řádku stačí jednoduchý příkaz, jak je zobrazeno na Obrázku 39. Tímto je zpřístupněná aplikace na předem nastavené IP adrese nebo doménovém jménu a portu. V našem případě se jedná o vývojové prostředí na adrese *http://127.0.0.1:5000/*.

```
root@lukas-VirtualBox:/home/lukas/Desktop/DP# python3 main.py
 * Serving Flask app "main" (lazy loading)
 * Environment: development
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 285-402-687
```

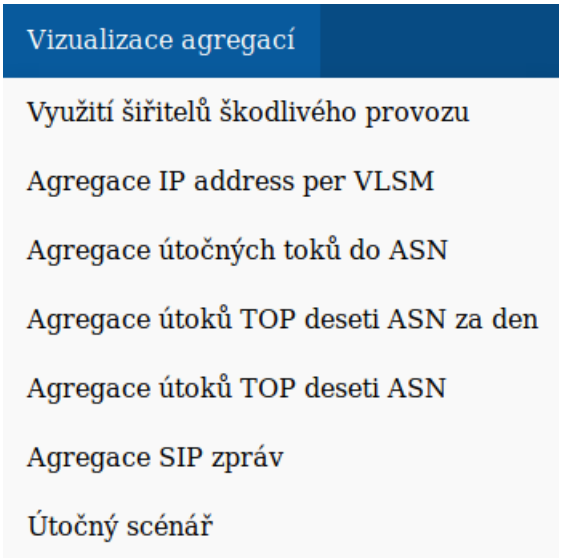
Obrázek 39: Příklad spuštění aplikace z terminálu.

Při navštívené této adresy se zobrazí domovská stránka aplikace, která je zobrazena na Obrázku 40.



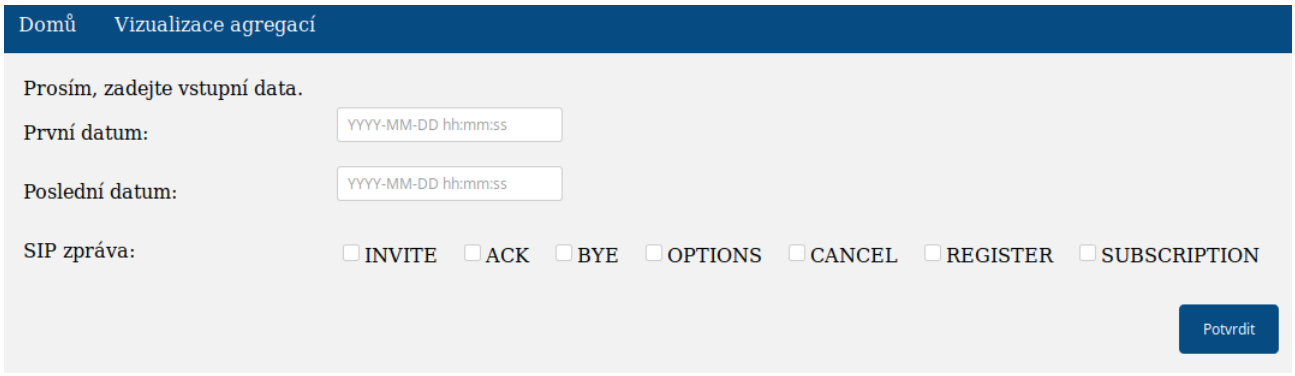
Obrázek 40: Domovská stránka webové aplikace.

Aplikace má jednoduchou navigační lištu se dvěma panely. První panel *Domů* vede na domovskou stránku aplikace. Druhý panel, *Vizualizace agregací*, je rozbalovací lišta, která slouží jako navigace k jednotlivým formulářům pro vizualizace. Rozbalovací lišta je zobrazena na Obrázku 41.



Obrázek 41: Rozbalovací lišta s odkazy na formuláře.

Všechny formuláře mají základní vstupní parametry pro zadání prvního a posledního data. Následující parametry záleží na typu vizualizace a agregace. Na Obrázku 42 lze vidět formulář pro agregaci založenou na signalizačním protokolu SIP. Zde byly přidány další vstupní parametry v podobě jednotlivých SIP zpráv. Uživatel tak může volit, které zprávy jej momentálně zajímají.



Obrázek 42: Rozbalovací lišta s odkazy na formuláře.

Po kliknutí na tlačítko *Potvrdit* se vstupní data odešlou nejprve k validaci, a pokud projdou validací, tak do funkcí řídících jednotlivé agregace a vizualizace. Uživateli jsou v aplikaci zobrazeny patřičné interaktivní grafy, jakmile jsou připraveny data pro vizualizaci. Může se stát, že uživatel zadá vstupní data, která jsou v rozporu s validátorem. V takovém případě se uživateli objeví chybová hláška a může se pokusit o zadání vstupních dat znovu.

Aplikaci byla implementována převážně jako ukázka možného rozhraní pro práci s agregacemi a vizualizacemi, a byla výhradně používána v lokálním prostředí. Není proto vhodná pro nasazení do produkčního prostředí.

5 Závěr

Cílem této diplomové práce bylo implementovat agregace informací o útocích na systém Beekeeper, které by umožnily detekci adresních rozsahů, z nichž útoky pocházejí, detekci DDoS útoků či charakteristických rysů jednotlivých nástrojů použitých k útokům.

Diplomovou práci jsem rozdělil do čtyř hlavních částí. První část diplomové práce se zabývá teoretickým úvodem do technologie VoIP. Ve druhé části diplomové práce jsem provedl návrh a analýzu možných agregací. Třetí část diplomové práce slouží k detailnímu popisu implementací jednotlivých agregací, jejich vizualizací a výkonnostního testování. Poslední část diplomové práce je věnována jednoduchému uživatelskému rozhraní.

V první části diplomové práce je uveden stručný úvod do technologie VoIP a to zejména úvod do signalizačního protokolu SIP. Je zde popsána architektura tohoto protokolu, adresování a přehled zpráv tak, jak je definován v RFC 3261. Je zde rovněž popsána bezpečnost protokolu SIP a bezpečnostní hrozby ve VoIP. V poslední podkapitole je stručný úvod do honeypotů ve VoIP.

Ve druhé části diplomové práce je popsána poskytnutá datová sada. Tato datová sada byla poskytnuta formou tabulky z databáze systému Beekeeper a obsahovala data v rozmezí od 18.12.2018 do 26.3.2019. Dále se zde hlavně zabývám analýzou této datové sady vzhledem k agregacím informací, které by mohly vézt k odhalení potenciálních DDoS útoků nebo charakteristických rysů nástrojů použitých k útokům. Na základě této analýzy byly vybrány celkem čtyři agregace. Pro účely těchto agregací bylo potřeba rozšířit databázový model Beekeeper o dvě další tabulky *bgp_info* a *ip_info*.

Třetí část diplomové práce se zabývá implementací jednotlivých agregací, jejich možných vizualizací a výkonnostním testováním. Každá z těchto agregací poskytuje jiný pohled na data z tabulky *mlp_attribute* a dají se z nich vyčíst poměrně zajímavé poznatky. Při analýze výsledků agregace dat na základě autonomních systému byly odhaleny jisté podobnosti a anomálie v datech u několika autonomních systémů. Další analyzovanou agregací byla agregace dat na základě šiřitelů škodlivého provozu. U této agregace vyšli najevo největší aktéři škodlivého provozu na úrovni autonomních systému a jednotlivých IP adres. Zde vyšlo najevo, že absolutně největším šiřitelem je AS2852, do něhož patří 14 IP adres, ze kterých pocházelo největší počet útoků. Třetí analyzovanou agregací byla agregace dat na základě dynamického útočného okna. Zde byly agregovány jednotlivé útoky do útočných toků na základě velikosti útočného okna. Tato agregace umožnila náhled na data v jiné podobě, než byly dosud zobrazeny. Dále pak agregace odhalila AS5167, jenž nebyl nejškodlivější coby do počtu útoků, ale byl nejrozšířenější vzhledem k počtu subnetů. Poslední implementovanou agregací byla agregace dat na základě signalizačního protokolu SIP. Tato agregace odhalila počty příchozích SIP zpráv. Takto můžeme vidět špičky jednotlivých zpráv, které se nápaditě podobají útokům typu zaplavování. Tyto špičky jsou nejvíc patrné u zpráv typu *register* a *options*. Díky této agregaci bylo možné zobrazení tzv. útočného scénáře, kde jednotlivé SIP zprávy byly agregovány do konkrétního autonomního sys-

tému. Velmi zajímavé výsledky poskytl tento scénář u AS2852, kde jde vidět několik současných výkyvů v datech u různých SIP zpráv. Díky těmto výkyvům můžeme být schopni zrekonstruovat tok celého útoku. Výkonnostní testování agregací odhalilo prostor pro optimalizaci jednotlivých algoritmů, jelikož některé agregace byly podstatně časově náročnější. Toto je však dáno také jakým způsobem jsou algoritmy navrženy, počtem dat potřebných k agregaci a počtem operací prováděných nad databází.

Ve čtvrté, poslední kapitole, je popsána webová aplikace, která byla navržena pro jednoduchou práci s agregacemi a vizualizacemi. Tato aplikace byl navržena jako jedna z možností práce s agregacemi, a je doporučeno používat ji výhradně v lokálním prostředí.

Všechny implementované agregace poskytují jiný náhled na data, jenž byla dodána. Díky těmto agregacím můžeme sledovat adresní rozsahy, autonomní systémy, útočné toky nebo jednotlivé SIP zprávy, které se podílí na škodlivém provozu. Takto jsme schopni určit zdroj útoku a sledovat, zdali se jedná o ojedinělou událost nebo opakující se. Takto připravená data se dají dále využít pro detekci DDoS útoků nebo detekci charatkteristických rysů útočných nástrojů, jako je např. SIPVicious.

Jisté agregační algoritmy nedosahují optimální rychlosti zpracování dat. Jedná se zejmáne o agregaci na základě dynamického útočného okna a vizualizaci na základě šiřitelů škodlivého provozu. Další místo pro optimalizaci se nachází u algoritmu, který agreguje zdroje útoků do uživatelem zadané VLSM. Webové rozhraní poskytuje jednoduchý způsob zobrazení vizualizací, avšak není vhodné pro produkční použití. Problémem tohoto rozhraní je dynamické zobrazení grafů TOP 10 ASN za den, kde počet grafů závisí na časovém rozpětí zadaným uživatelem.

Literatura

[1] University of Southern California. *Internet Protocol* [online]. Říjen 1981 [cit. 2020-05-11]. Request for Comments (RFC) 791. Dostupné z: https://tools.ietf.org/html/rfc791

[2] ROSENBERG J., SCHULZRINNE H., CAMARILLO G., JOHNSTON A., PETERSON J., SPARKS R., HANDLEY M., SCHOOLER E. *SIP: Session Initiation Protocol* [online]. Červen 2002 [cit. 2020-05-11]. Request for Comments (RFC) 3261. Dostupné z: https://tools.ietf.org/html/rfc3261

[3] LEVIN O. *H.323 Uniform Resource Locator (URL) Scheme Registration* [online]. Duben 2003 [cit. 2020-05-11]. Request for Comments (RFC) 3508. Dostupné z: https://tools.ietf.org/html/rfc3508

[4] SCHULZRINNE H., CASNER S., FREDERICK R, JACOBSON V. *RTP: A Transport Protocol for Real-Time Applications* [online]. Červenec 2003 [cit. 2020-05-11]. Request for Comments (RFC) 3350. Dostupné z: https://tools.ietf.org/html/rfc3550

[5] SHEKH-YUSEF R., AHRENS D., BREMER S. *HTTP Digest Access Authentication* [online]. Září 2015 [cit. 2020-05-11]. Request for Comments (RFC) 7616. Dostupné z: https://tools.ietf.org/html/rfc7616

[6] POSTEL J. *User Datagram Protocol* [online]. Srpen 1980 [cit. 2020-05-11]. Request for Comments (RFC) 768. Dostupné z: https://tools.ietf.org/rfc/rfc768.txt

[7] BERNERS-LEE T., FIELDING R., MASINTER L. *Uniform Resource Identifiers (URI): Generic Syntax* [online]. Srpen 1998 [cit. 2020-05-11]. Request for Comments (RFC) 2396. Dostupné z: https://www.ietf.org/rfc/rfc2396.txt

[8] FIELDING R., GETTYS J., MOGUL J., FRYSTYK H., MASINTER L., LEACH P., BERNERS-LEE T. *Hypertext Transfer Protocol – HTTP/1.1* [online]. Červen 1999 [cit. 2020-05-11]. Request for Comments (RFC) 2616. Dostupné z: https://tools.ietf.org/html/rfc2616

[9] POSTEL J., REYNOLDS J. *FILE TRANSFER PROTOCOL (FTP)* [online]. Říjen 1958 [cit. 2020-05-11]. Request for Comments (RFC) 959. Dostupné z: https://tools.ietf.org/html/rfc959

[10] BAUGHER M., MCGREW D., NASLUND M., CARRARA E., NORRMAN K. *The Secure Real-time Transport Protocol (SRTP)* [online]. Březen 2004 [cit. 2020-05-11]. Request for Comments (RFC) 3711. Dostupné z: https://tools.ietf.org/html/rfc3711

[11] RESCORLA E. *The Transport Layer Security (TLS) Protocol Version 1.3* [online]. Srpen 2018 [cit. 2020-05-11]. Request for Comments (RFC) 3711. Dostupné z: https://tools.ietf.org/html/rfc8446

[12] DAIGLE L. *WHOIS Protocol Specification* [online]. Září 2003 [cit. 2020-05-11]. Request for Comments (RFC) 3912. Dostupné z: https://tools.ietf.org/html/rfc3912

[13] SCHMIDT, Holger, Chi-Tai DANG a Franz J. HAUCK. Proxy-based Security for the Session Initiation Protocol (SIP). In: *2007 Second International Conference on Systems and Networks Communications (ICSNC 2007)* [online]. IEEE, 2007, 2007, s. 42-42 [cit. 2020-05-11]. DOI: 10.1109/ICSNC.2007.64. ISBN 0-7695-2938-0. Dostupné z: http://ieeexplore.ieee.org/document/4300014/

[14] KEROMYTIS, Angelos D. A Comprehensive Survey of Voice over IP Security Research. *IEEE Communications Surveys & Tutorials* [online]. 2012, **14**(2), 514-537 [cit. 2020-05-11]. DOI: 10.1109/SURV.2011.031611.00112. ISSN 1553-877X. Dostupné z: http://ieeexplore.ieee.org/document/5742777/

[15] VoIP Security Alliance. *VoIP Security and Privacy Threat Taxonomy"* [online]. 2005.10.24. Dostupné z: http://www.voipsa.org/Activities/VOIPSA_Threat_Taxonomy_0.1.pdf

[16] XIN Jianqiang. *Security Issues and Countermeasure for VoIP*. Dostupné z: https://www.sans.org/reading-room/whitepapers/voip/security-issues-countermeasure-voip-1701

[17] PATEL, Mayank a BUDDHDEV Bharat. "Analysis of Security Threats in Voice Over Internet Protocol (VOIP)." *Control Theory and Informatics 3* (2013): pp. 30-37. [online] [cit. 2020-05-11] Dostupné z: <https://www.iiste.org/Journals/index.php/CTI/article/view/8148>

[18] DAKUR, Aditya a DAKUR, Shruthi. "Eavesdropping and interception security hole and its solution over VoIP service." 2014 *IEEE Global Conference on Wireless Computing & Networking (GCWCN)* (2014): pp. 6-10.[online] [cit. 2020-05-11]. Dostupné z: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7030837&tag=1>

[19] ŘEZÁČ F., VOZŇÁK, M., ROZHON, J. Bezpečnost v komunikacích, VŠB-TUO: 2. rozšířené vydání,185 str., 2013

[20] The Honeynet-Project: "Know Your Enemy: Honeypots", [online] (2002) [cit. 2020-05-11]. Dostupné z: < http://www.sane.nl/events/sane2002/papers/honeynet.PDF >

[21] ALBASHIR, Anas Abd Almonim Nour. Detecting unknown vulnerabilities using Honeynet. In: *2015 First International Conference on Anti-Cybercrime (ICACC)* [online]. IEEE, 2015, 2015, s. 1-4 [cit. 2020-05-11]. DOI: 10.1109/Anti-Cybercrime.2015.7351929. ISBN 978-1-4799-7620-1. Dostupné z: < http://ieeexplore.ieee.org/document/7351929/>

[22] FAN, Wenjun, Zhihui DU a David FERNANDEZ. Taxonomy of honeynet solutions. In: *2015 SAI Intelligent Systems Conference (IntelliSys)* [online]. IEEE, 2015, 2015, s. 1002-1009 [cit.

2020-05-11]. DOI: 10.1109/IntelliSys.2015.7361266. ISBN 978-1-4673-7606-8. Dostupné z: <http://ieeexplore.ieee.org/document/7361266/>

[23] ŠAFAŘÍK, Jakub. *Distribuovaný systém klasifikace útoků pro VoIP infrastrukturu využívající protokol SIP* [online]. Ostrava, 2016 [cit. 2020-05-11]. Disertační práce. Vysoká škola báňská - Technická univerzita Ostrava. Dostupné z: <https://dspace.vsb.cz/handle/10084/116856>

[24] DENNY CZEJDO, Bogdan, Erik M. FERRAGUT, John R. GOODALL a Jason LASKA. Network Intrusion Detection and Visualization Using Aggregations in a Cyber Security Data Warehouse. *International Journal of Communications, Network and System Sciences* [online]. 2012, 05(09), 593-602 [cit. 2020-05-11]. DOI: 10.4236/ijcns.2012.529069. ISSN 1913-3715. Dostupné z: <http://www.scirp.org/journal/doi.aspx?DOI=10.4236/ijcns.2012.529069>

A Dokumentace pro práci s výsledným řešením

Aplikace je logicky rozdělena do dvou částí: backendová část a frontendová část. Pro implementaci backendové části byl vybrán programovací jazyk Python ve verzi 3.7.1. Frontendová část je pak rozdělena na části napsané ve značkovacím jazyce HTML a skritpovacím jazyce JavaScript.

A.1 Backendová část

Backendová část se skládá celkem ze tří souborů:

- main.py - obsahuje Flask aplikaci pro řízení frontendu, včetně validací a směrování mezi odkazy,
- bgp_aggregation.py - obsahuje základní agregaci na základě autonomních systémů, včetně podpůrných funkcí,
- data_export.py - obsahuje veškeré funkce pro export dat pro vizualizační algoritmy, včetně některých agregací.

Tyto soubory by měly být umístěny ve stejném adresáři. Je třeba dát si pozor přístupová práva, pod kterými se soubory spouští, kvůli přístupu do adresářů jako je např. */var/lib/mysql-files/*. Aplikace neprovádí kontrolu, zda li je spuštěna databázová služba, nýbrž se předpokládá, že služba běží a všechny tabulky jsou vytvořeny.

A.1.1 Soubor main.py

Obsahuje hlavní funkci pro vytvoření instance třídy Flask, který vytvoří jednoduchý HTTP server na adrese *http://127.0.0.1:5000/*. Je možné veřejně zpřístupnit server na lokální síti přidáním parametru *host="0.0.0.0"*, nicméně toto nastavení velmi nedoporučuji, jelikož uživatelský frontend byl vytvořen pouze jako ukázka možnosti ovládání celé aplikace a tudíž postrádá nezbytné bezpečnostní mechanismy. Soubor main.py obsahuje několik funkcí a Flask cest. V souboru jsou také importovány soubory *bgp_aggregation.py* a *data_export.py* pro volání jednotlivých agregačních funkcí a funkcí pro export dat.

A.1.2 Soubor bgp_aggregation.py

Obsahuje algortimy pro agregaci dat na základě autonomních systémů. Hlavní funkce tohoto souboru je *aggregate_to_asns*, která postupně spouští ostatní funkce pro výpočet agregace. Soubor také obsahuje několik podpůrných funkcí jakou jsou *convert_to_binary*, *compare_bits* a *write_output*. Funkce *aggregate_to_asns* je volána v souboru main.py. V průběhu této agregace se stahují externí soubory do složky **WWW**, která by měla být umístěna v kořenovém adresáři souboru *bgp_aggregation*.

A.1.3 Soubor data_export.py

Soubor obsahuje veškeré funkce pro export dat včetně několika agregačních funkcí. Jednotlivé funkce jsou volány v souboru main.py na základě uživatelského vstupu. Mezi klíčové funkce patří *aggregate_attack_streams* a *aggregate_sip_messages*. Datové soubory jsou exportovány do složky **data**, která by měla být umístěna v kořenovém adresáři souboru data_export.py.

A.2 Frontendová část

Tato část se skládá z několika HTML, JS a CSS souborů. HTML soubory jsou umístěny ve složce **templates** v kořenovém adresáři backendových souborů. JS a CSS soubory jsou umístěny ve složce **static** ve stejném kořenovém adresáři. Mezi jednotlivé HTML soubory patří:

- index.html
- data_asn.html
- data_cidr.html
- data_scenario.html
- data_attack_source_usage.html
- data_data_sip_messages.html
- data_top_ten.html
- data_top_ten_asn_per_day_dynamic.html

Sobor *index.html* slouží jako domovská stránka webové aplikace. Ostatní soubory slouží k zobrazení formuláže pro konkrétní vizualizace a následné zobrazení vizualizace. Mezi JS a CSS soubory patří:

- style.css
- data_top_ten.css
- index_jquery.js
- chart_outer_jquery.js

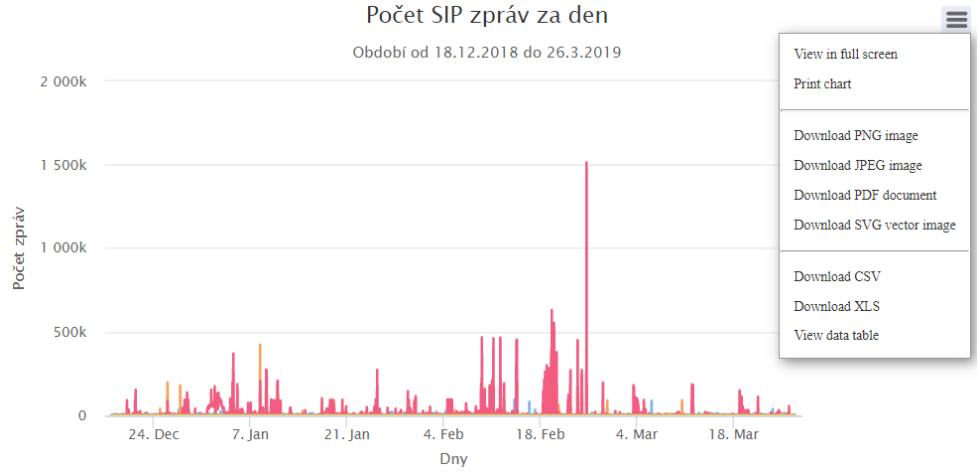
CSS soubory slouží pro načtení stylů v HTML souborech. Soubor *style.css* se využívá ve všech souborech, zatímco soubor *data_top_ten.css* se využívá pouze ve dvou. Soubory .JS se využívají pro načtení JavaScriptového kódu pro vykonání akce ve formuláři a dynamickému zobrazování a odebírání HTML *div* bloků. Soubor *index_jquery.js* slouží pouze pro souboru *index.html*, zatímco soubor *chart_outer_jquery.js* slouží pro ostatní HTML soubory.

A.3 Práce s interaktivními grafy

Pro jednodušší práci s grafy byly přidány mechanismy jako jsou přepínání mezi lineární a logaritmickou osou nebo vypnutí a zapnutí zobrazení jednotlivých křivek. Tyto mechanismy jsou popsány níže.

A.3.1 Tlačítko exporting

U většiny grafů bylo implementováno tlačítko, které má několik různých funkcí. Mezi tyto funkce patří zobrazení grafu na celou obrazovku nebo vytištění grafu, stažení grafu ve formátech PNG, JPEG, PDF a SVG a zobrazení nebo stažení tabulky s daty ve formátu CSV nebo XLS. Tlačítko je umístěno v pravém horním rohu grafu a je zobrazeno na Obrázku 43.



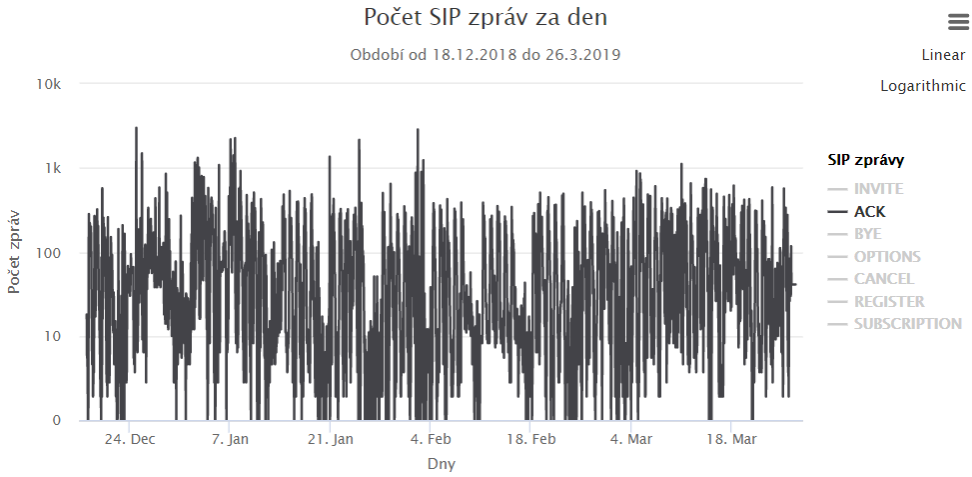
Obrázek 43: Detail na graf s tlačítkem "exporting".

A.3.2 Přepínání mezi měřítky

U některých grafů lze data interpretovat ve dvou měřítkách - lineárním a logaritmickém. Přepnutí na jiné měřítko lze kliknutím na tlačítko *Linear* nebo *Logarithmic*. Tyto tlačítka jsou umístěna v pravém horním rohu grafu pod tlačítkem exporting A.3.1. Výchozí měřítko grafů je lineární.

A.3.3 Vypnutí/zapnutí zobrazení křívek

Grafy s legendou mají možnost vypnutí nebo zapnutí jednotlivých křivek. To provedeme po kliknutí na konkrétní křivku v legendě umístěné v pravé části grafu. Takto může uživatel zobrazit křivky, které jsou pro něj zajímavé. Příklad je uveden na Obrázku 44, kde je zaplá pouze křivka ACK.



Obrázek 44: Ukázka grafu s vypnutými křivkami.

A.3.4 Přiblížení na ose x

Při vykreslování velkého počtu dat je možné, že graf bude méně čitelný. Z tohoto důvodu byl přidán mechanismus pro přiblížení části grafu na ose x. Toto provedeme stisknutím levého tlačítka myši, táhnutím pro označení oblasti a následným puštěním tlačítka. Po přiblížení se v pravém horním rohu grafu zobrazí tlačítko *Reset zoom* pro přechod do původní podoby. Takto přiblížený graf můžeme vidět na Obrázku 24.

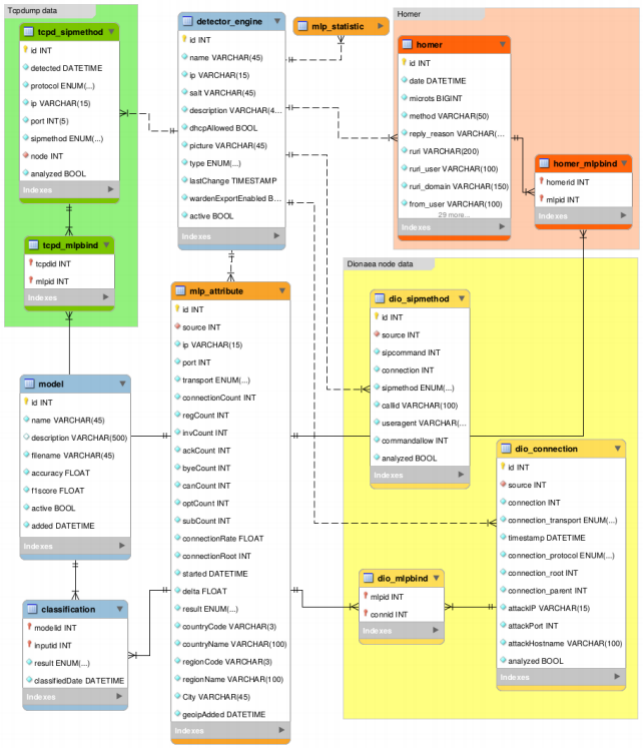
A.3.5 Detail na vnitřní prstenec grafu

U paprskového grafu je možnost zobrazení detailu vnitřního prstence. Toto provedeme kliknutím na vnitřní oblast, jejíž detail si přejeme vidět. Prstencový graf lze vidět na Obrázku 25, detail na jednu z vnitřních oblastí na Obrázku 26.

A.3.6 Detail na druhou úroveň stromového grafu

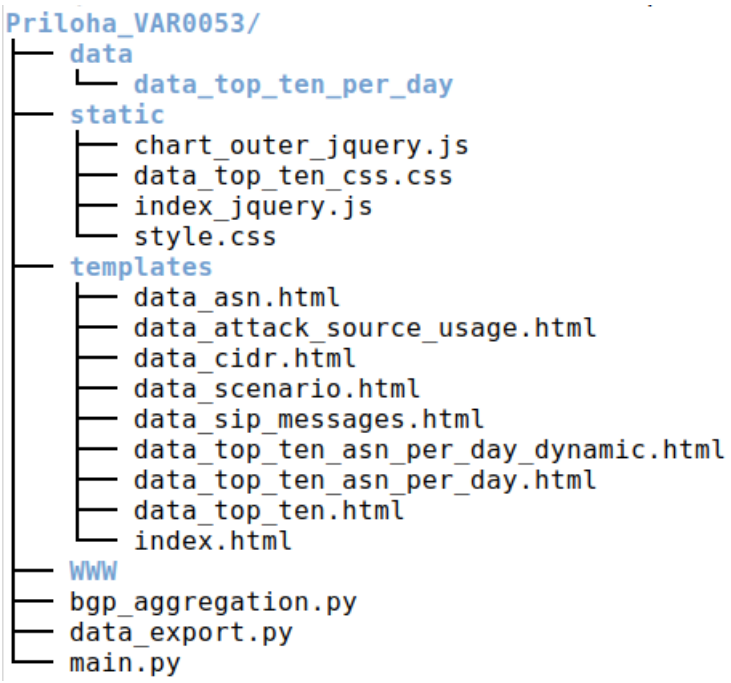
Stromové grafy jsou velmi specifické svým tvarem a mohou být víceúrovňové. Pro účely této diplomové práce stačily stromové grafy se dvěma úrovněmi. Po kliknutí na konkrétní oblast z první úrovně přejdeme do úrovně druhé. Detail na druhou úroveň lze vidět na Obrázku 32.

B Obrázky



Obrázek 45: Struktura MySQL databáze pro Beekeeper (převzato z [23])

C Příloha v IS Edison



Obrázek 46: Příloha v IS Edison.